Towards Semantic Trajectory Knowledge Discovery

Luis Otavio Alvares[1,2], Vania Bogorny[1], Bart Kuijpers[1], Bart Moelans[1], Jose Antonio Fernandes de Macedo[3], Andrey Tietbohl Palma[2]

[1] Theoretical Computer Science,
Hasselt University & Transnational University of Limburg, Belgium

[2]Instituto de Informática – UFRGS, Brazil

[3]Ecole Polytechnique Fédérale de Lausanne, Switzerland

Abstract. Trajectory data play a fundamental role to an increasing number of applications, such as transportation management, urban planning and tourism. Trajectory data are normally available as sample points. However, for many applications, meaningful patterns cannot be extracted from sample points without considering the background geographic information. In this paper we propose a novel framework for semantic trajectory knowledge discovery. We propose to integrate trajectory sample points to the geographic information which is relevant to the application. Therefore, we extract the most important parts of trajectories, which are *stops* and *moves*, before applying data mining methods. Empirically we show the application and usability of our approach.

1. Introduction

Trajectory data are normally obtained from location-aware devices that capture the position of an object at a specific time interval. The collection of these kind of data is becoming more common, and as a result large amounts of trajectory data are available in the format of sample points. In many application domains, such as transportation management, animal migration, and tourism, useful knowledge about moving behavior or moving patterns can only be extracted from trajectories if the background geographic information where trajectories are located is considered. Therefore, there is a necessity for a special processing on trajectory data before applying data mining techniques.

An example which expresses such necessity is shown in Fig. 1. In Fig. 1 (left) we can visualize a set of trajectories, that apparently have no meaning. In Fig. 1 (right) we have the same trajectories over the geographic space, where we can visually infer the geographic location (Paris) and the intersection of trajectories with touristic places (e.g. Eiffel tower) and hotels.

Raw trajectory data are collected from the Earth's surface similarly to any kind of geographic data (see raw data level in Fig. 2). It is known that raw geographic data require a lot of work to be transformed into maps, normally stored into shape files (processed data level in Fig. 2). The processed geographic data are generated for any application domain, and therefore are *application independent*. They can be used, for instance, to build *geographic databases* of applications of transportation management, tourism, urban planning, etc (application level in Fig. 2). Geographic databases, on the other hand, are *application dependent*, and therefore, will contain only the entities of processed geographic data that are relevant to the application.

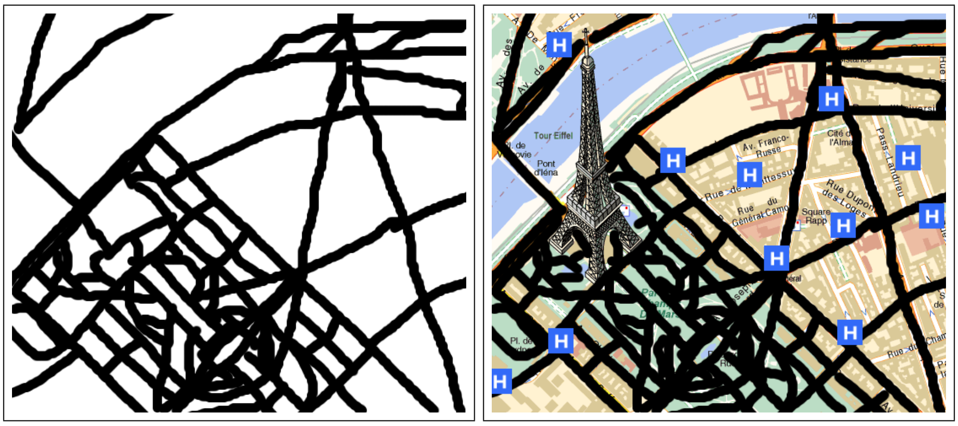

Figure 1. (left) trajectories and (right) trajectories with geographic information

Data mining and knowledge discovery are on the top level, and are also application dependent. In any data mining task the user is interested in patterns about a specific problem or application. For instance, transportation managers are interested in patterns about traffic jams, crowded roads, accidents, etc, but are not interested in, for instance, patterns of animal migration.

In trajectory pattern mining, however, mining has been basically applied over raw trajectory data or trajectory sample points, as shown in Fig. 2. Indeed, these algorithms as far as we know have not considered the background geographic information in the mining process. We claim that meaningful patterns for decision making processes in real applications cannot be extracted from raw trajectory data. For data mining and knowledge discovery, trajectories represented as sample points need to be a priori integrated with the background geographic information which is relevant to the application, for then apply data mining methods. This a priori integration will generate semantic trajectory patterns, that facilitate the user's task to analyze and interpret the knowledge in post-processing steps.

In this paper, we propose a general framework to integrate geographic data with trajectories in the form of sample points, in order to create semantic trajectories for knowledge discovery. Therefore, we will consider trajectories as a set of *stops* and *moves*, where stops are the important places for the application, defined by the user, and moves are transitions between consecutive stops.

The remaining of the paper is organized as follows: Sect. 2 presents the related works and the main contributions. In Sect. 3 we introduce the basic concepts about trajectories, stops, and moves. In Sect. 4 we present a framework to integrate geographic data with trajectories in order to create a semantic trajectory database for knowledge discovery. In Sect. 5 we present some application examples to show the usability of our approach. In Sect. 6 we conclude the paper and present some directions of future work.

2. Related Work and Contribution

Several data models have been proposed for efficiently querying raw trajectory data such as [Kuijpers and Othman 2007, Wolfson et al. 1998], but only a few approaches [Brakatsoulas et al. 2004, Spaccapietra et al. 2007, Mouza and Rigaux 2005]

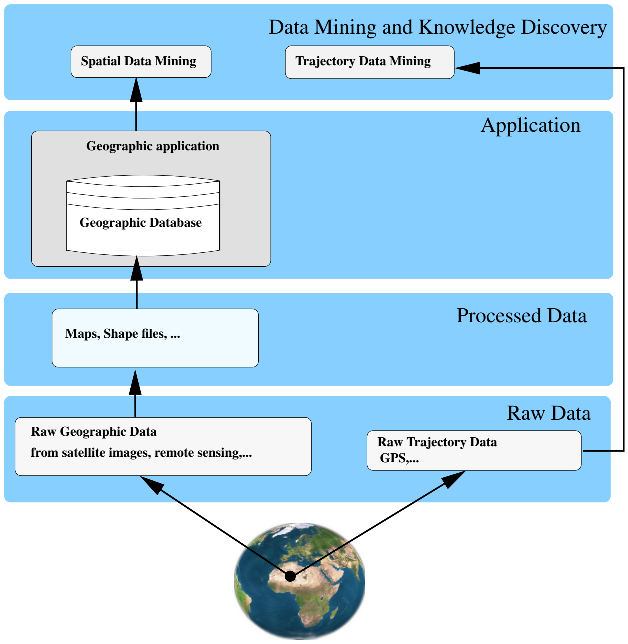

Figure 2. Current framework of spatial and trajectory knowledge discovery

consider semantics and background geographic information.

In [Wolfson et al. 1998] the main focus relies on the geometric properties of trajectories. In [Mouza and Rigaux 2005] moving patterns are extracted from data by defining the patterns a priori. For instance, find all trajectories that move from a zone A to a zone B and cross zone Z. In this work, moving patterns are the trajectories that follow a given pattern. In our work we will extract moving patterns from data which are not known a priori and which are frequent in a minimal number of trajectories.

In [Brakatsoulas et al. 2004] a semantic model for trajectories has been proposed as well as relationships of trajectories with the background geographic information. This model, however, is restricted to a specific domain, where the road network is the geographic information that is taken into account, similarly to [Güting et al. 2006].

In this paper we adopt the semantic model presented by [Spaccapietra et al. 2007], which introduces the concept of stops and moves, and which we have extended in [Alvares et al. 2007a] with trajectory pattern relationships using data mining and reverse engineering. In [Alvares et al. 2007b] we have shown how the semantic trajectory data model can facilitate the formulation of trajectory queries and reduce the time complexity.

From the data mining perspective, several trajectory mining algorithms have been proposed in recent years [Cao et al. 2006, Gudmundsson and van Kreveld 2006, Laube et al. 2005, Lee et al. 2007, Li et al. 2004, Nanni and Pedreschi 2006, Tsoukatos and Gunopulos 2001, Verhein and Chawla 2006]. Although some approaches consider regions of interest, the semantics and the geographic information behind trajectories has not been considered. Some approaches find dense patterns, where moving objects are in the same region, close to each other, and move in the same direction [Cao et al. 2006]. Other approaches find long patterns [Tsoukatos and Gunopulos 2001] and flock patterns [Gudmundsson and van Kreveld 2006], but short patterns like moves between two places [Verhein and Chawla 2006] may be useful for some applications to understand moving behavior.

Since most existing trajectory data mining approaches deal with raw trajectories, we propose a continued study of our previous work [Alvares et al. 2007b] of trajectory queries for semantic trajectory data mining and knowledge discovery.

In general the main contributions of this paper include: (i) a framework for semantic trajectory data mining and knowledge discovery that is performed only once, to transform raw trajectories into semantic trajectories; (ii) an analysis with application examples to show the applicability of the proposed framework over different classical data mining methods.

3. Stops and Moves of a Trajectory

In this section we present the concepts which are necessary to represent semantic trajectories, using stops and moves to integrate geographic information to raw trajectories, defined in [Alvares et al. 2007b].

Definition 1 A *sample trajectory* is a list of space-time points $\langle(x_0, y_0, t_0), (x_1, y_1, t_1), \ldots, (x_N, y_N, t_N)\rangle$, where $x_i, y_i, t_i \in \mathbb{R}$ for $i = 0, \ldots, N$ and $t_0 < t_1 < \cdots < t_N$.

To transform trajectory sample points in *stops* and *moves*, it is necessary to provide the important places of the trajectory which are relevant for the application. The important places correspond to different spatial feature types [OGC 1999], specified in a geographic database. For each relevant spatial feature type that is important for the application, a minimal amount of time is specified by the user, such that a trajectory should continuously intersect this feature in order to be considered a stop. This pair is called candidate stop.

Definition 2 A *candidate stop* C is a tuple (R_C, Δ_C), where R_C is a (topologically closed) polygon in $\mathbb{R}^2$ and Δ_C is a strictly positive real number. The set R_C is called the *geometry* of the candidate stop and Δ_C is called its *minimum time duration*.

An *application* $\mathcal{A}$ is a finite set $\{C_1 = (R_{C_1}, \Delta_{C_1}), \ldots, C_N = (R_{C_N}, \Delta_{C_N})\}$ of candidate stops with mutually non-overlapping geometries $R_{C_1}, \ldots, R_{C_N}$.

In case that a candidate stop is a point or a polyline, a polygonal buffer is generated around this object, and thus represent it as a polygon in the application, in order to overcome spatial uncertainty.

Definition 3 Let T be a trajectory and let

$$\mathcal{A} = (\{C_1 = (R_{C_1}, \Delta_{C_1}), \ldots, C_N = (R_{C_N}, \Delta_{C_N})\})$$

be an application.

Suppose we have a subtrajectory $\langle(x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), \ldots, (x_{i+\ell}, y_{i+\ell}, t_{i+\ell})\rangle$ of T, where there is a (R_{C_k}, Δ_{C_k}) in $\mathcal{A}$ such that $\forall j \in [i, i+\ell] : (x_j, y_j) \in R_{C_k}$ and $|t_{i+\ell} - t_i| \geq \Delta_{C_k}$, and this subtrajectory is maximal (with respect to these two conditions), then we define the tuple $(R_{C_k}, t_i, t_{i+\ell})$ as a *stop of T with respect to $\mathcal{A}$*.

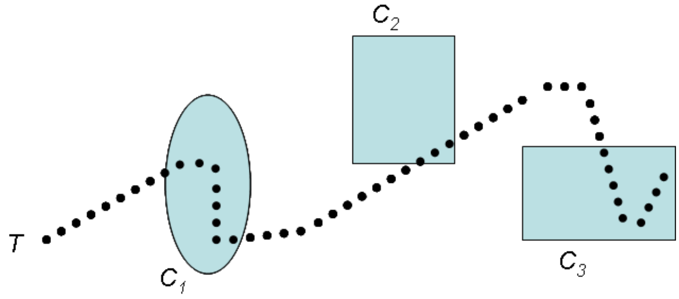

Figure 3. Example of application with three candidate stops

Definition 4 A *move* of a trajectory T is: (i) a subtrajectory of T delimited by two temporally consecutive stops of T; OR (ii) the subtrajectory of T between the starting point of T and the first stop of T; OR (iii) the subtrajectory of T between the last stop of T and the ending point of T; OR (iv) the trajectory T itself, if T has no stops.

Figure 3 illustrates these concepts. In this example, there are three candidate stops C_1, C_2 and C_3 with geometries G_1, G_2 and G_3 respectively. Let us consider that the spatial projection of the trajectory T is from left to right. First, T is outside any candidate stop, so we start with a move. Then T enters G_1. Since the duration of staying inside G_1 is long enough, C_1 constitutes the first stop of T. Next, T enters G_2, but the time duration is shorter than D_2, so this is not a stop. We therefore have a move until T enters G_3 which fulfills the requests to be a stop. The trajectory T ends with the stop C_3.

4. The Proposed Framework

Figure 4 presents the proposed framework to extract stops and moves from raw trajectories in order to create a semantic trajectory database for data mining and knowledge discovery. At the bottom layer are the raw trajectory data, collected in a specific region (e.g. city, state, country). For this same region there are topographic maps with different layers of geographic information, that were originally developed for *any* application domain. These data can be used to build geographic applications and semantic trajectory databases.

Geographic data are preprocessed and made available for any application. Similarly, raw trajectory data should be preprocessed as well, in order to remove noise, identify gaps, etc. The processed trajectory data can be used in different applications. However, to be useful, these data need to be integrated with the geographic information which is relevant to a trajectory application, in order to obtain semantic trajectories. For this integration, the candidate stops can be either extracted from the processed data like shape files, or from a geographic database, when the trajectory application is related to an existing geographic application (e.g. transportation management).

In our framework, in the application level, semantic trajectory data are created with stops and moves, where the user inform the relevant spatial feature types(e.g. airport, hotel) and their minimal duration. In [Alvares et al. 2007b] we present an algorithm to find stops and moves. In general words, the algorithm verifies for each point of a trajectory T if it intersects a candidate stop C. In affirmative case, the algorithm looks if the duration of the intersection is at least equal to a given threshold D. If this is the case, the intersected candidate stop is considered as a stop, and this stop is recorded. In Table 1(a) is an

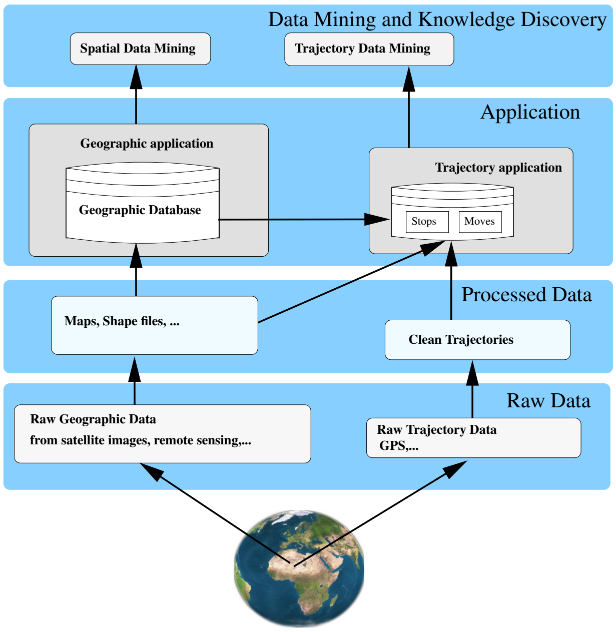

Figure 4. Proposed framework for semantic trajectory knowledge discovery

example of a stop dataset where the attribute *gid* corresponds to the instance of the spatial feature type (stopType) where this stop occurs (e.g. airport, hotel).

In case a relevant spatial feature type is a point or a polyline, we generate a polygonal buffer around this object, and thus represent this object as a polygon in the application. This is important to avoid precision problems.

A move is recorded between the previous stop and the latest one. This previous stop can be null, if the latest stop is the first stop of the trajectory. When a move is inserted into the set of moves, its geometry is also added. For some applications, it might be interesting to know the spatial features that a move has crossed, and therefore we keep the geometry of the move for further spatial analysis, which are out of the scope of this work.

Table 1(b) shows an example of a move dataset. Notice that in this dataset instead of storing the identifier of the stop (Sid) for normalization, we keep the instance and the name of the spatial feature type (Gid and stopType). We also keep the time in the move dataset. While in databases data are supposed to be normalized, for data mining and knowledge discovery data are unnormalized in one single table or one single file, because of the restrict input format of most data mining algorithms. The storage of the spatial feature type identifier and name will significantly facilitate both querying and mining directly over the move dataset, as we will explain in the following section.

Stops and moves are stored in two relations in the geographic database, being already integrated to the background geographic information that is relevant to the application. Therefore, the patterns extracted from the semantic trajectories will refer to a specific application. An important remark which can be noticed in the stops and moves dataset is that the identifiers of both stops and moves represent the order as they have occurred in the trajectory. This will facilitate the extraction of patterns in which the order

Table 1. Example of Stops and Moves Dataset

(a) Stops

Sid	Tid	Gid	stopType	enterTime	leaveTime
1	1	1	Airport	09:45	10:05
2	1	2	Hotel	11:10	14:08
3	1	1	TouristPlace	14:30	16:03
4	1	8	ShoppingArea	17:00	19:03
5	1	1	Hotel	19:05	21:00
1	2	1	Hotel	7:00	7:50
2	2	2	TouristicPlace	8:32	11:04
3	2	1	TouristicPlace	13:15	15:05
4	2	8	ShoppingArea	16:15	18:08
…	…	…	…	…	…

(b) Moves

Mid	Tid	Gid1	stopType1	Gid2	stopType2	geometry	startTime	endTime
1	1	1	Airport	2	Hotel	LINESTRING(…)	10:05	11:10
2	1	2	Hotel	1	TouristicPlace	LINESTRING(…)	14:08	14:30
3	1	1	TouristicPlace	8	ShoppingArea	LINESTRING(…)	16:03	17:00
4	1	8	ShoppingArea	1	Hotel	LINESTRING(…)	19:03	19:05
1	2	1	Hotel	2	TouristicPlace	LINESTRING(…)	07:50	08:32
2	2	2	TouristicPlace	1	TouristicPlace	LINESTRING(…)	11:04	13:15
3	2	1	TouristPlace	8	ShoppingArea	LINESTRING(…)	15:05	15:15
…	…	…	…	…	…	…	…	…

is important, like sequential pattern mining.

The process of generating stops and moves from trajectories is normally performed only once, while data mining algorithms may be applied to these data several times until interesting patterns may be found.

According to the kind of knowledge the user is interested in, different data mining algorithms can be applied to the dataset of stops and moves, such as frequent patterns, association rules, and sequential patterns. To illustrate the applicability of semantic trajectories for mining, the following section presents some application examples of the proposed framework.

5. Application Examples and Framework Implementation

Hereafter we present examples of a tourism application, considering the trajectories of visitors in a touristic city, and present an overview of the implementation of the proposed framework into the data mining toolkit Weka [Frank et al. 2005] and the PostGIS database.

5.1. Semantic Trajectory Data Analysis

Let us consider that there is a dataset of trajectories represented as sample points, which already passed through a cleaning process, representing the trajectories of tourists in Paris in July and August. Part of this dataset is shown in Table 2(a). Besides the trajectory data of visitors, there is also geographic information of the city. Let us suppose that for this application, the interesting feature types (candidate stops) include hotel, touristic place, shopping area, train station, and airport. Samples of Hotel and Touristic Place are shown in Table 2(c) and 2(b).

Considering the given relevant spatial feature types, we apply only once the algorithm to extract stops and moves from the trajectory sample points. An example of stops and moves dataset is shown in Table 1. Now let us compare the trajectory dataset in Table 2(a), that has no semantic information, and the stops and moves examples shown in Table 1. First, over raw trajectories, basically all queries will need a spatial join operation, either for the analysis of the pure trajectories or for the integration with geographic information. Second, only geometric patterns can be directly inferred over the trajectory sample points. Over the stops and moves, on the other hand, geographic information was added in one step, and several non-spatial queries and data mining tasks can be directly applied. By just looking at the data (stops and moves) we can infer some knowledge like people arriving at the airport, going from hotels to touristic places, visiting touristic places and going to shopping area, and so on.

Several different kinds of patterns can be of interest for the user of a tourism application, and we will show how easily they can be obtained from semantic trajectories (stops and moves).

Let us consider that the user of the tourism application would like to answer some specific questions about the movement of tourists in the city, starting with questions like Q1. Question Q1 is a typical frequent pattern mining problem, and can be answered by mining the stops dataset.

Q1: *Which are the places most frequently visited by tourists in the morning?*

The answer can be on either instance or type granularity, according to the objective of the application. For instance, if the user is interested in patterns about specific places, like Ibis Hotel, Eiffel Tower, then the attributes *stopType* and *gid* have to be considered in the mining task. If the interest is in more general patterns (e.g. Hotel to Touristic Place), then the attribute *stopType* is enough. For single patterns including only one stop, a simple query that counts the occurrence of stops in the different trajectories where the visitors have stopped and where the *enterTime* is before 12AM, will answer this question. However, if the interest is in longer patterns, then a classical non-spatial frequent pattern mining algorithm can be applied over the stops dataset. Considering the stops relation as an example, some possible frequent patterns could be those shown in A1. In A1(a), the frequent stops represent a pattern that expresses that in 10% of the trajectories, tourists that stop at *Eiffel Tower* do also stop at *Orsay*. The second pattern A1(b) expresses a more general pattern, that 80% of the trajectories that have a stop at a touristic place also have a stop at a hotel.

A1: (a) $\{TouristicPlace_1, TouristicPlace_3\}$ (s=10%),
(b) $\{TouristicPlace, Hotel\}$ (s=80%)

Several queries may be in mind of a tourism application user about the relationships of places visited by tourists. Question Q2 shows a simple example of such query:

Q2: Is there any relation between visited touristic places and hotels?

To answer this query, the association rule mining technique can be applied over the stops data set. Association rules like those shown in A2 can be generated. The first rule A2(a) expresses that tourists that stop at Ibis hotel do also stop at Eiffel tower. This happens in 20% of the trajectories, and with a confidence of 70%. The second rule expresses that 50% of the tourists that stop at the Louvre museum and at the Invalids, do also stop at Hilton hotel. The support of this rule is 25%.

Table 2. (a) Raw trajectories, (b) Touristic Place and (c) Hotel

(a) Trajectory

Tid	geometry	timest
1	POINT(48.890018 2.246100)	08:25
1	POINT(48.890018 2.246100	08:26
...	...	...
1	POINT(48.890020 2.246102)	08:40
1	POINT(48.888880 2.248208)	08:41
1	POINT(48.885732 2.255031)	08:42
...	...	...
2	POINT(48.858434 2.336105)	09:04
2	POINT(48.853611 2.349190)	09:05
...	...	...

(b) Touristic Place

gid	Type	Name	Geometry
1	Monument	Eiffel Tower	POINT(48.858330 2.294333)
2	Museum	Louvre	POINT(48.862220 2.335556)
3	Museum	Orsay	POINT(48.860000 2.326944)
4	Museum	Invalides	POINT(48.855555 2.312778)
5	Church	Notre Dame	POINT(48.853611 2.349167)
6	Church	Saint Chapelle	POINT(48.855545 2.345003)
...	...	...	...

(c) Hotel

gid	Name	Geometry
1	Ibis	POINT(48.890015 2.246110)
2	Hilton	POINT(48.880005 2.283889)
...	...	...

A2: (a) $\{Hotel_1\} \Rightarrow \{TouristicPlace_1\}$ (s=20%) (c=70%)
(b) $\{TouristicPlace_2\} \wedge \{TouristicPlace_4\} \Rightarrow \{Hotel_2\}$ (s=25%) (c=50%)

For several reasons like the modification of existing bus routes or the creation of new shuttle bus routes, questions like Q3 may be interesting for the user. Question 3 can be answered with the sequential pattern mining task over both stops and moves dataset. For instance, sequential patterns like A3 can be extracted from the stops dataset. In A3(a) visitors that go to the Louvre (TouristicPlace 2) in the morning, go to the Notre Dame Church in the afternoon (TouristicPlace 5).

Q3: *Which is the sequence of touristic places most frequently visited and when these visits occur?*

A3: (a) $\{[Louvre]_{morning}, [NotreDame]_{afternoon}\}$ (s=8%)
(b) $\{[EiffelTower]_{morning}, [Invalides]_{morning}, [NotreDame]_{afternoon}\}$ (s=6%)

We can also apply the sequential pattern mining technique over the move data set. In this case, longer pattern are generated, and lower minimum support should be considered, since at least 4 stops are part of the shortest pattern, with only 2 moves. In A4(a) the example shown that trajectories that have a move from Orsay museum to Eiffel tower, also have a move from the Invalides to Notre Dame church, in this order. A longer pattern is shown in A4(b), where the trajectories that have a move from Louvre to Notre Dame in the morning, do also have a move from Notre Dame to Saint Chapelle in the Afternoon and then from Saint Chapelle to Pompidou Center, in this order.

A4: (a) $\{[Orsay - EiffelTower], [Invalides - NotreDame]\}$ (s=5%)
(b) $\{[Louvre - NotreDame]_{morning}, [NotreDame - SaintChapelle]_{afternoon}, [SaintChapelle - PompidouCenter]_{afternoon}\}$ (s=1%)

Once trajectory samples have been transformed into semantic trajectories, any classical data mining task can be applied to discovery of different patterns. Different analysis can be performed over these data, avoiding several spatial joins that do not need to be computed after trajectories have been integrated with the relevant geographic information.

The query examples shown in this section are simple and have the objective to illustrate the applicability of the proposed framework and how the user of the application is benefited with this approach. The framework has been implemented into a data mining toolkit, as a data preprocessing method to prepare trajectories for data mining. Because of space limitations only a brief overview of the tool is present in the following section.

5.2. Implementation: Weka-STPM

Weka-STPM (Semantic Trajectory Preprocessing Module) is a preprocessing module to automatically generate semantic trajectories. Weka-STPM is a new graphical GUI for spatio-temporal data preprocessing, similar to the Weka-GDPM [Bogorny et al. 2006], which has been developed for automatic geographic data preprocessing for data mining.

Weka-STPM connects to a geographic database stored in PostGIS. In Weka-STPM the user may clean trajectories in order to generate the processed trajectory data, represented in the processed data level of our framework. This step is to remove noise, to specify the trajectory identifier and to transform timestamps if necessary. After this step, the user can choose the spatial feature types from the geographic database that are relevant to the application, i.e., the candidate stops. The minimal time duration for a relevant feature be considered a stop is also provided.

In one step the clean trajectory samples are integrated with the relevant spatial feature types, and stops and moves are generated as two relations in the geographic database. Two arff files, which is the input format required by Weka are generated. The user can choose the granularity level of the stops and moves: feature instance or feature type. The time intervals can also be generated at several granularities according to the user specification. For instance: (a) journey period: *morning*, *afternoon*, and *evening*, rush hours: (a) [07:00-09:00], (b) [17:00-19:00]; weekdays and weekend, etc. This automatic discretization and transformation functions are the most useful preprocessing operations for trajectory data mining.

After this step has been performed, the user can apply any classical data mining technique available in Weka.

6. Conclusions and Future Work

In this paper we have shown that knowledge discovery is a process that is application dependent, and therefore there is need to integrate geographic information into trajectories in order to extract more meaningful patterns. In this paper we have presented a framework to perform this integration, in which we consider the background geographic information that is relevant for the application. In our framework we integrate the geographic information and trajectories in a data preprocessing step, before apply data mining techniques.

Without using such kind of conceptualization it would be hard or maybe impossible to understand moving behavior in trajectories in a specific application. On the one hand, the proposed framework may need some computational time to extract stops and moves from trajectory data, although this step is performed only once. On he other hand, this a priori integration reduces the query complexity [Alvares et al. 2007b] and will significantly reduce the time for mining trajectories and interpreting the patterns.

The main contribution of this paper is for the application user, that will need much less time to understand the patterns, since the resultant semantic trajectories are in some way summarized and interpreted.

For some applications it might be interesting to extract stops and moves without giving a priori the relevant feature types. Indeed, some interesting trajectory patterns may be related to spatial feature types that the user may not know a priori and that may not be discovered when the user has to provide the relevant features. To address this point, the future ongoing work is the use of the clustering technique to find possible stops that are not known a priori by the user.

Acknowledgments

This research has been funded by the Brazilian agencies CAPES and CNPq, and the European Union (FP6-IST-FET programme, Project n. FP6-14915, GeoPKDD: Geographic Privacy-Aware Knowledge Discovery and Delivery)

References

Alvares, L. O., Bogorny, V., de Macedo, J. F., Moelans, B., and Spaccapietra, S. (2007a). Dynamic modeling of trajectory patterns using data mining and reverse engineering. In *Twenty-Sixth International Conference on Conceptual Modeling - ER2007 - Tutorials, Posters, Panels and Industrial Contributions*, volume 83. CRPIT.

Alvares, L. O., Bogorny, V., Kuijpers, B., de Macedo, J. A. F., Moelans, B., and Vaisman, A. (2007b). A model for enriching trajectories with semantic geographical information. In *ACM-GIS*, New York, NY, USA. ACM Press.

Bogorny, V., Palma, A. T., Engel, P., and Alvares, L. O. (2006). Weka-gdpm: Integrating classical data mining toolkit to geographic information systems. In *WAAMD*, pages 9–16. SBC.

Brakatsoulas, S., Pfoser, D., and Tryfona, N. (2004). Modeling, storing, and mining moving object databases. In *IDEAS '04: Proceedings of the International Database Engineering and Applications Symposium (IDEAS'04)*, pages 68–77, Washington, DC, USA. IEEE Computer Society.

Cao, H., Mamoulis, N., and Cheung, D. W. (2006). Discovery of collocation episodes in spatiotemporal data. In *ICDM*, pages 823–827. IEEE Computer Society.

Frank, E., Hall, M. A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., and Trigg, L. (2005). Weka - a machine learning workbench for data mining. In Maimon, O. and Rokach, L., editors, *The Data Mining and Knowledge Discovery Handbook*, pages 1305–1314. Springer.

Gudmundsson, J. and van Kreveld, M. (2006). Computing longest duration flocks in trajectory data. In *GIS'06: Proceedings of the 14th annual ACM International Symposium on Advances in Geographic Information Systems*, pages 35–42, New York, NY, USA. ACM Press.

Güting, R. H., de Almeida, V. T., and Ding, Z. (2006). Modeling and querying moving objects in networks. *VLDB J.*, 15(2):165–190.

Kuijpers, B. and Othman, W. (2007). Trajectory databases: Data models, uncertainty and complete query languages. In *ICDT*, pages 224–238.

Laube, P., Imfeld, S., and Weibel, R. (2005). Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19(6):639–668.

Lee, J., Han, J., and Whang, K. Y. (2007). Trajectory clustering: A partition-and-group framework. In *SCM SIGMOD International Conference on Management Data (SIGMOD'07)*, Beijing, China.

Li, Y., Han, J., and Yang, J. (2004). Clustering moving objects. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 617–622, New York, NY, USA. ACM Press.

Mouza, C. and Rigaux, P. (2005). Mobility patterns. *GeoInformatica*, 9:297–319(23).

Nanni, M. and Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *J. Intell. Inf. Syst.*, 27(3):267–289.

OGC (1999). Topic 5, opengis abstract specification - features (version 4) (1999). Available at: http://www.OpenGIS.org/techno/specs.htm. Accessed in August (2005).

Spaccapietra, S., Parent, C., Damiani, M.-L., de Macedo, J. A. F., Porto, F., and Vangenot, C. (2007). A conceptual view on trajectories. Technical report, Ecole Polytechnique Federal de Lausanne.

Tsoukatos, I. and Gunopulos, D. (2001). Efficient mining of spatiotemporal patterns. In Jensen, C. S., Schneider, M., Seeger, B., and Tsotras, V. J., editors, *SSTD*, volume 2121 of *Lecture Notes in Computer Science*, pages 425–442. Springer.

Verhein, F. and Chawla, S. (2006). Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In Lee, M.-L., Tan, K.-L., and Wuwongse, V., editors, *DASFAA*, volume 3882 of *Lecture Notes in Computer Science*, pages 187–201. Springer.

Wolfson, O., Xu, B., Chamberlain, S., and Jiang, L. (1998). Moving objects databases: Issues and solutions. In Rafanelli, M. and Jarke, M., editors, *SSDBM*, pages 111–122. IEEE Computer Society.