Where you stop is who you are: understanding people's activities by places visited

Laura Spinsanti,[1] Fabrizio Celli,[1], Chiara Renso[2]
[1]*EPFL, Lausanne, Switzerland*
[b]*KDDLab, ISTI, CNR, Pisa, Italy*

Abstract. The increasing availability of people traces - collected by portable devices - poses new possibilities and challenges for the study of people mobile behaviour. However, the raw data produced by such portable devices is poor from a semantic point of view, thus the gap between the person's activity and the raw collected data generated by the activity is still too wide. The work presented in this paper aims to define an algorithm to understand the activity of a moving person from the sequence of places she visited. The contribution is twofold. On one hand, an algorithm to associate each stop of the traveling person to a list of probable visited places is introduced. On the other hand, the obtained sequence of places is classified into a possible activity performed by the moving person. Preliminary experimental results on a dataset of people moving by car in the city of Milan are reported.

Keywords. GPS trajectory, behaviour inference from GPS, Points of Interest.

Introduction

The last decade has seen mobile communications technologies pervading our society. Mobile wearable tracking devices sense the movement of people and vehicles, generating large volumes of mobility data, which represent the traces of people's activity.

The interest in developing formal frameworks for understanding people activity dates back to many decades ago. However, only in the recent years, with the increasing availabilities of movement datasets collected from GSM or GPS equipped devices, we have the possibility of studying people activities from their movement traces. Nowadays, several application areas would benefit from an extensive study on people's activities such as traffic management, public transportation, commercials and advertising, security and police, hazard evacuation management, location based services and so on.

Despite the fact that data collected from mobile devices is increasing its location accuracy, it is not improving in the same way their quality in terms of semantic richness. This means that the semantic gap between raw data collected from mobile devices and the personal activity that generated the traces is still too wide to be filled

[1] Corresponding Author: laura.spinsanti@epfl.ch.

by current technologies. As a consequence, techniques to semantically enrich the collected data are necessary to (semi-) automatically infer the person's activity given her/his location traces.

The approach presented in this paper aims at enriching people's movements, represented as trajectories, with semantic information about the places visited during her/his travel. The basic assumption is that people stop, during the movement, to reach a goal. In this context, the sequence of places visited by a person during her/his movement tells us a lot about the activity she/he is performing, so that we can infer, with a degree of approximation, which is the behaviour of the moving person during the analyzed movement. For example, a person visiting museums, restaurants, monuments and eventually ending the day in a hotel can be associated to a tourist activity. To do that, we first need to identify the places visited by people during their trips; secondly, we need to associate these places to an activity typically performed in those places; thirdly, we want to infer the (probably) overall behaviour associated to a trip by analyzing the specific activities carried on during people's stops.

In the current approach we assume that a tracking device is installed into a vehicle (e.g. a car). Then, we identify the stops of the trajectory and we associate the places probably visited by the tracked person. More in detail, we propose an algorithm to associate each stop in a user's trajectory to a list of possible visited places and we associate to each of these places a probability. Finally, depending on the kinds of activities associated to the identified place, the trajectory is classified into a (probable) trajectory behaviour.

The paper is structured as follows. Section 1 reports some related work, Section 2 introduces the basic definitions and assumptions of the approach and describes the details of the conceptual model. Sections 3 and 4 give the details of the algorithm. The experimental results are reported in Section 5, and conclusions and future work are stated in Section 6.

1. Related Work

The work proposed in this paper is essentially based on the pioneering work of Spaccapietra et al. in [8] where authors propose a conceptual model for *semantic trajectories*. Trajectories are defined as a time-space function that record the changing of the position of an object moving in space during a given time interval. Semantic trajectories are defined as sequences of *stops* (where the moving object stays still during a time interval) and *moves* (the part of a trajectory where the position of the object changes). All stops are temporally disjoint, while each move is delimited by two consecutive stops. The basic assumption behind the notion of stop is that the place where a person stops is of some interest for her/him. Therefore, each stop is somehow associated to a *Place of Interest* (POI).

Analysis methods on semantic trajectories have been proposed in [4], where the authors propose methods to compute stops from raw GPS trajectories. In a later work they propose mining methods to analyze semantic trajectories such as frequent pattern and association rules [5]. The association between a POI and a trajectory stop has been done considering the places of interest of the application and they consider the spatially closer POI. However, they do not explicitly consider the temporal validity of the association (i.e. if the POI exists or it is accessible during the actual stop), neither the probability value associated to each stop–POI pair. Furthermore, they use data mining to find common patterns among trajectories, while we focus on the semantic interpretation of each individual trajectory.

The identification of behaviour of tracked people is not new in the literature. The system Athena proposed in [5] classifies semantic trajectories and mined patterns into behaviour categories. Here, ontologies have been exploited to represent domain knowledge, and axioms are used to interpret the features (such as stops) of a given trajectory/pattern as behaviour (e.g. people stopping in hotels and museums are classified as tourists). The stop is not a point, but a cell in a grid and no explicit stop–POI association has been computed.

The AIDA system proposed by [1] analyses the movement behaviour of car drivers in order to identify the set of goals the driver would like to achieve in his/her trip. Furthermore, AIDA involves an understanding of the city incorporating context information such as business and shopping districts, tourist and residential areas, as well as real-time event information and environmental conditions. Driver preferences are also integrated into AIDA. One mandatory task for AIDA is to predict the destination of the driver as well as the most likely route that he/she will follow. This will in turn allow for useful reactions from AIDA such as proposing route alternatives when something unexpected happens in the predicted route, or providing the right information at the right time (e.g. a fuel warning before passing by a gas station) or even helping save energy. However, they do not explicitly face the problem of inferring the visited POI, instead they focus on mathematical models to predict the destination of a driver.

The approach in [9] aims at automatically inferring the transportation modes from trajectories recorded by personal GPS devices, as a step towards recognizing human behaviour and understanding. Their approach is based on a "change point-based segmentation" to effectively partition a trajectory into segments of transportation modes maintaining the segment as long as possible. The method infers the transportation mode using Basic Features such as velocity and acceleration but it is improved by the use of Advanced Features such as heading change rate stop rate and velocity change rate which consistently improves the basic method. However, combining the change point segmentation method and the Decision Tree classification further improves the accuracy of results. A further post-processing graph-based method mines an implied graph containing the commonsense constraint of the real world and typical user behaviours.

Andrienko and Andrienko in [2] propose *Visual Analytics* methods to infer semantics from raw trajectories. They focus the sources and destinations of trajectories on the basis of the temporal criterion, i.e. according to the time spent in a location. Compared to the present approach they do not infer an overall activity from the detected POI nether they compute a probability list associated to each POI.

Kifer and Stein [7] propose a method for user intention recognition in the mobile case. They propose a framework where movement information through GPS data is used by a system of production rules and classification technique for the intention recognition process. They use a grammatical formalism with spatial knowledge. Despite the final objective is somehow similar to ours, this approach mainly focuses on movement features such as speed, angles etc. to segment a trajectory, whereas our approach relies on the stop where no signal have been detected to infer the visited POIs and consequently infer the user activity.

The novelty of our approach, with respect to [5], [1] and [9], is manifold. Indeed, since our focus is based on a real case scenario, we take into account many spatial and temporal aspects to realistically associate a stop to POIs. Furthermore, we explicitly build a probability ranking list of possible visited places, whereas other approaches just choose one place (typically the spatially closest). Moreover, the methodology proposed here to associate a possible POI to a stop explicitly considers the temporal dimension, taking into account the nearest temporally-reachable place, whereas other approaches only consider the spatial dimension, therefore the closeness of the POI to the stop. Also, the proposed algorithm, besides computing the stop–POI association, classifies each trajectory into a behavioural class. The difference with the work in [5] is that in that work the classification has been done by an ontology with encoded predefined behavioural rules. In our work, the classification of people behaviour is first inferred by the probability measures of each POI and then further refined by user-defined rules.

2. OVERVIEW OF THE APPROACH

We assume the moving object is a person that travels using a transportation means associated to a traceable (GPS-) device (car, bus, metro, train). The person gets out of the transportation mean to reach the final destination walking. During this time interval the person is not traceable. We are assuming that the person stopped in a place since she/he is interested in visiting that place, so the geographical objects that could represent the goal of the stop are called places of interest. A *Place of Interest*, or POI, is a (urban) geo-referenced object where a person may carry out a specific activity. Each stop may be associated to one or more POIs.

The approach presented here is based on the analysis of the POI visited by the user during a stop, disregarding the actual movement between the stops (the “moves”). Therefore, our semantic trajectory representation is limited to the sequence of stops.

Our scenario represents a real life situation in an urban environment, where a person moves to reach a specific place in which she/he performs a specific activity. In particular, we model a scenario when a person drives as close as possible to her/his goal, then parks the car and walks to her/his place of interest. This means that the person is visiting some of the *reachable* places around, not necessary the closest. We also have to consider the temporal dimension to avoid impossible matches. For example, the closest POI can have opening and closing time and when the stop occurs during the POI closing time, this should be disregarded. In this approach, it is possible to associate to a stop several POIs and, in turn, associate to each POI several activities. For example, in a shopping mall area it is possible to associate to a stop both a supermarket, a postal service, a cinema, a fast food, and so on. For each of these places, it is possible to associate several activities to perform in like shopping, eating, working. Furthermore, different activities can be characterized by different temporal durations. For instance, a stop in a supermarket for 30 minutes is probably associated to a shopping activity, whereas a stop lasting more than 4 hours, is probably corresponding to a working activity. This is formalized in a conceptual model, presented in the following section.

2.1. The Conceptual Model

The conceptual relationship between trajectories, stops, POIs and activities is illustrated in the conceptual model of Figure 1. The diagram has been inspired by the conceptual model presented in [8] and represents the main concepts of the trajectory behaviour approach presented in this paper. Here, a *trajectory* is a spatio-temporal concept composed by BES (Begin, End, Stop). BES are connected by moves. The Point Of Interest (POI) concept is modeled as a spatio-temporal entity. Implementing this model means that trivial common sense constraints, like the fact that a person can't go to a POI during the closing time or days, are embedded in the data specification as trigger constraints. In fact, only during the opening interval the association with the stop is possible. The entity *AssociatedP&A* is a ternary relationship that links each BES to a POI and to the activities that can be performed in the POI. For example, a stop at a restaurant can be associated to an "eating" activity. An activity (or a pattern of activities) may in turn be associated to a behaviour, which is associated to the trajectory concept, thus representing the fact that a trajectory expresses a behaviour through the performed activities.

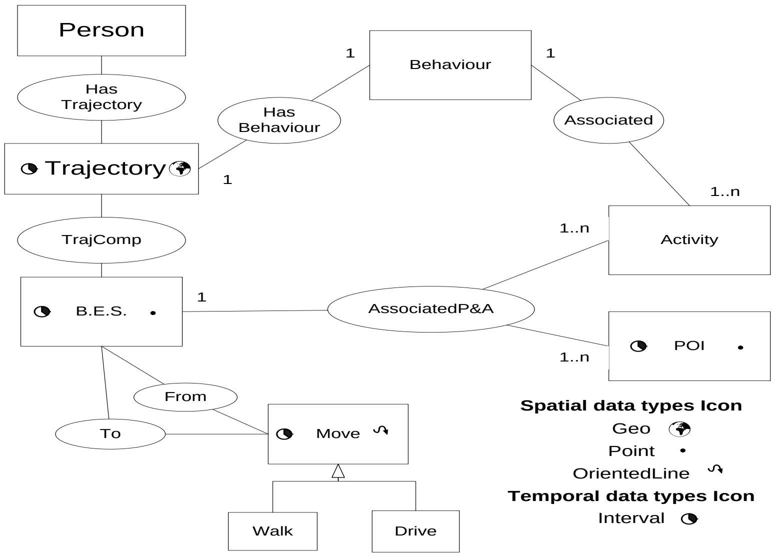

Figure 1 The trajectory behaviour conceptual model

2.2. The approach

The methodology to infer the activity performed by moving users is depicted in Figure 2. Here, movement data is collected from tracking devices and trajectories are reconstructed from them (for examples of trajectory reconstruction techniques see [6]). Given the trajectories, the first step is to identify the stops of the trajectories [3, 8]. The stops are the portion of trajectories where the movement stops for a given time duration and where we assume the user is performing some activity. During this step we disregard all the stops that cannot be associated to "interesting" places, such as the stops with a very short time duration that are typical of the movement itself, such as a traffic light. The component "Visited POI" takes as input the list of stops and the POIs and computes a ranked list – based on probabilities – of possible points of interests visited by the user. This component is implemented by an algorithm called "*AssignToPOI*" presented in details later. The ranked list becomes the input of the component *TrajectoryBehaviour* which performs an inference on the probable beheviour expressed by the user during her/his trip, based on the sequence of visited POIs. As a result, a trajectory is tagged with the most probable behaviour associated to that particular trip.

In next sections we illustrate the details of the two modules namely "*Vistited POIs*" and "*TrajectoryBehaviour"*.

We assume that each POI is associated to a predefined category and categories, in turn, are organized in a conceptual hierarchy represented as "*is_a*" relationship in an ontology of POIs. Figure 3 shows an example of ontology of POIs categories, namely

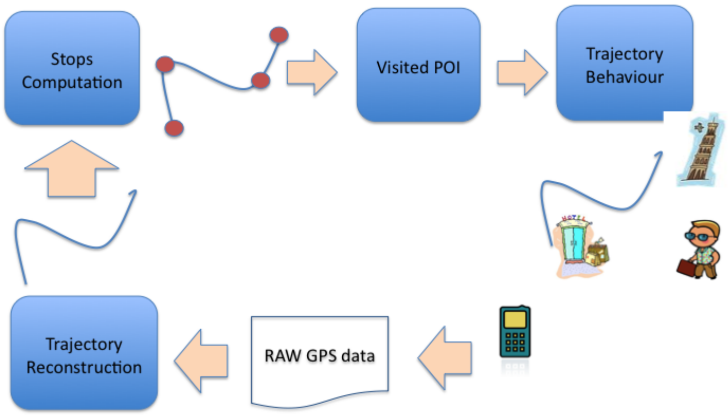

Figure 2 The schema of the approach

tourist place, work place, entertainment place. The base bottom level represents the specific POI category such as museum, hotel, etc. Each of the POIs category may belong to more than one "super-category". In the example of Figure 3, the eating place POI category is both a kind of tourist place and working place expressing the fact that a restaurant of fast food may be a kind of work place for people working there (the cook, the waiter etc), or a tourist place to represent the fact that typically restaurants are attended by tourists, and so on. The categories have associated ontology properties that describe the semantic information about the place, such as the average visit time, opening and closing time. This information can be exploited by the semantic rules in the *visitedPOI* module to give constraints or simply refine the POI assigned to a stop.

Similarly, activities are organized in a taxonomy which generalizes the kinds of activities of interest for the movement analysis. For example, the "tourist" activity can be specialized in "family tourist activity", and so on.

These two taxonomies are organized in an ontology as depicted by Figure 3. We have a relationship between places and activities, according to the conceptual model, stating that an activity is typically performed in a place. The ontology contains additional contextual information about the POIs, such as the closing time, the average visit time, and it is used in the *VisitedPOI* and *TrajectoryBehaviour* components, presented in the next sections.

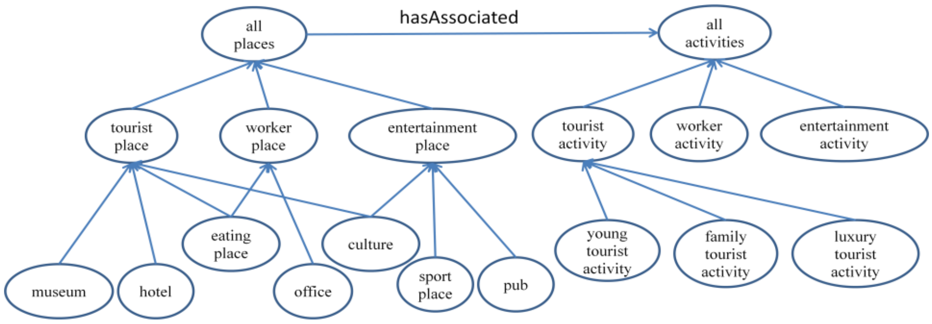

Figure 3 The ontology of POI and Activity

3. DETECTING THE VISITED POI

The objective of this module is to compute the points of interests visited by moving people. Indeed, having the places visited by a user is the basis to understand the activity performed by a user during her/his movement and this, in turn, in semantically characterizing the overall behaviour of the tracked person.

In the next section we present the main steps of the *AssignToPOI* algorithm, followed by the semantic rules employed by the algorithm to infer the most probable places. These rules are of two kinds: the *constraint rules* aimed at excluding some "impossible" POIs based on commonsense constraints, and the *probability rules* aimed at adjusting the probability of the assigned POIs based on domain knowledge.

3.1. The AssignToPOI Algorithm

The algorithm is based on the definition of semantic trajectory T as a sequence of stops, where a stop S is a triple:

$$Trajectory = \langle S_1, \ldots, S_N \rangle \qquad S_i = \{\langle x_i, y_i \rangle, T\} \qquad i = 1..N$$

where $\langle x_i, y_i \rangle$ represents the geographical position of the stop, T represents the duration of the stop. A POI is a triple:

$$P_x = \{\langle x_x, y_x \rangle, Cat_x, TM_x\}$$

where $\langle x_x, y_x \rangle$ represents the geographical position of the POI – assuming that it is a point -, Cat_x is the category and TM_x is the average-visit-time of the POI. *Cat* is the category of the POI in the places taxonomy.

The input parameters of the algorithm are:

- a set of semantic trajectories
- a set of POIs
- the taxonomy of places categories *Cat*
- the average speed of a walking person *v* (that is an application dependant parameter).

We assume the following symbol:

- $\boldsymbol{S_X}$ is the current *stop*
- $\boldsymbol{T_X}$ is the *stop* duration
- $\boldsymbol{TM_i}$ is the *average- visit-time* of the *POI* P_i
- $\boldsymbol{d_{i,X}}$ **=distance(S_X,P_i)** is the distance between the *POI* P_i and the *stop* S_X

- $\boldsymbol{TP_{i,x}} = \mathbf{d_{i,x}}/\boldsymbol{v}$ is the time a person needs to cover the distance $d_{i,x}$
- $\mathbf{TE_{i,x}=T_X\text{-}2\cdot TP_{i,x}}$ is the maximum time that a person has to visit the *POI* P_i, that is to say the difference between the duration of the stop S_X and the time needed to reach P_i and to come back to the stop.
- L_X is the list of POIs associated to S_X

The output of the algorithm is a probability measure $P_{i,x}$ associated to each pair of stop S_i and POI P_x, of analyzed trajectories.

The algorithm can be described in four different steps that compute probabilities by successive refinements. Let us consider the generic trajectory T belonging to the input set of trajectories:

Step 1 – Selecting POIs. For each stop of a trajectory T, we compute the set of POIs that can be associated to it. Two conditions are taken into account: (1) having enough time to go and come back from stop to POI and (2) having enough time to visit the POI. This means that the amount of time a person could spent in a place is not the complete stop duration, but the time needed to cover the distance between the POI and the stop must be taken into account. Moreover, the distance is assumed to be the *walking* distance over a road network. In work [9], authors propose an algorithm that maps a trajectory on a road map with an accuracy of 92%, when the measurement error is no more than 8 meters. Therefore stops and POIs are mapped over a road map and an algorithm to compute the minimum distance is applied (the Dijkstra Algorithm). It is worth observing that, in general, the longer is the duration of the stop, the higher is the number of POIs associated to that stop. Furthermore we can assume that a person walks for no more than X meters and for Y seconds, being X and Y application dependent variables. Optionally, if the resulting set is too large, an upper bound can be defined. Analogously, places categories can be used to limit the number of POIs for each category.

This step runs *offline* since it is a pre-processing operation applied before the algorithm runs. More formally,

$$
\begin{aligned}
&L_X = \langle\,\rangle \\
&\forall P_i \in POI, \qquad if\ TE_{i,x} > 0\ AND\ TM_i \le TE_{i,x} \\
&then\ L_X = \langle L_X, P_i\rangle
\end{aligned}
$$

Step 2. Assign Probability to POI. In this step we assign a probability to all POIs of all stops of trajectory T on the basis of their distances from the associated stop and of their average-visit-time: thus, the POI nearest to the stop will have more probability to be its goal. This probability is then refined by the comparison between the average-visit-time of the POI and the duration of the stop. In other words, since we assume that for each stop we have only one POI visited, we prefer a POI whose average-visit-time is closer to the duration of the stop. When constraints rules are available, they are applied at this step (see par 3.2).

We call ***SpatP*** the spatial probability, linked to distance, and ***TempP*** the temporal probability, linked to time spent. The formula that computes the probability related to the distance between the *POI* P_i and its associated stop is:

$$SpatP_{i,x} = \frac{\frac{\sum_k d_{i,k}}{d_{i,x}}}{\sum_k \frac{\sum_k d_{i,k}}{d_{i,k}}}$$

The closer is the *POI* to the *stop*, the higher is the value returned by this formula.

On the other hand, the formula that takes into account the *average- visit-time* of the *POI* P_i is the following:

$$TempP_{i,x} = \frac{\frac{TM_x}{TE_{i,x}}}{\sum_x \left(\frac{TM_x}{TE_{i,x}}\right)}$$

that, as we have just explained, prefers the *POI* whose average-visit-time is closer to the duration of the stop. We can use these two formulas to compute the total probability:

$$P_{i,x} = \frac{TempP_{i,x} + \alpha \times SpatP_{i,x}}{\alpha + 1}$$

where α is a weight used to give more or less importance to the distance-criterion.

Step 3 - Updating probability using past history. Previous steps assigned a probability for each POI of each stop. We can refine these probabilities by considering the *previous* stops of the trajectory and their places categories. The basic assumption is that when a high number of stops of the trajectory T we associated POIs belonging to the same places category C, it is probable that the current stop will belong to the same category. For example, we assume that a person who visited a lot of POIs of "tourist" category will probably visit other tourist places. Obviously the drawback of this heuristic is that is can be biased by people "randomly" moving between different POI categories, such as visiting a museum, going to shop and going to work. In the assumption that most of the movements are "uniform", this heuristic can be useful to refine the POI probability.

However, we use this heuristics only when step 2 identifies a POI with a probability that don't exceed a given threshold T_1 (that is an application dependent parameter). A POI whose probability is too low is considered an "uncertain" POI. Therefore, we analyze the history of the current trajectory to update the probabilities of the POIs of the current stop.

For this step we need a threshold T_1, that represents a threshold for uncertain stops. For example, we can choose $T_1 = 0.5 + \frac{0.5}{size(\mathrm{L_i})}$, that means that we perform a past analysis only when we have a *POI* with a probability smaller than 50%.

$$\mathrm{P(Cat_x)} = \sum_{k=1}^{i} P_{k,y} \quad where\ Cat_y = \mathrm{Cat_x}$$

$$if\ \forall\ P_x \in L_i\ P_{i,x} < T_1\ then$$

$$if\ \#\mathrm{L_i} = 1 \text{ then } P_{i,x} = \mathrm{P(Cat_x)}$$

$$\text{else } P_{i,x} = \frac{P_{i,x}}{\sum_k \mathrm{P(Cat_k)}} \cdot \mathrm{P(Cat_x)}$$

To better clarify this step let us consider an example. We have a taxonomy of place categories with three categories B, D, and G. For the first stop we have only one *POI* associated to category D, with a probability of 100%. For the second stop the situation has been reported in Table 1.

POI	CATEGORY	PROBABILITY
P1	B	10%
P2	G	25%
P3	G	15%
P4	D	50%

Table 1. Probability assigned to stop 2

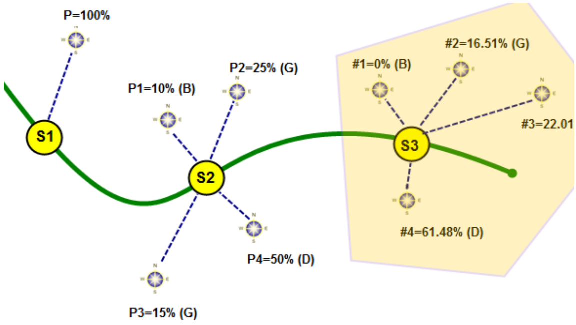

Figure 4 The schema of the approach

Here, we have 10% for category B, 40% for category G and 50% for category D. Note that POI P2 is assigned to the category G with a ratio of 25/40, while P3 of 15/40 (where 40 is the percentage of the category G computed at the second stop). For stop S3, we have four *POIs* associated (see Figure 4), where #2 and #3 belong to the category G.

So at the third stop, for the category G we have an aggregate probability of $16.51 + 22.01 = 38.52$. We can update probabilities of current *categories* on the basis of the categories of the past stops:

$$P(G3) = \frac{0+40+38.52}{3} = 26.18$$

$$P(D3) = \frac{100+50+61.48}{3} = 70{,}49$$

$$P(B3) = \frac{0 + 10 + 0}{3} = 3.33$$

So the probabilities for current *POIs* will be:

$$P_{T2} = \frac{16.51}{38.52} * 26.18 = 11.22\% \qquad P_{T3} = \frac{22.01}{38.52} * 26.18 = 19.96\%$$

$$\boldsymbol{P_{T4} = 70.49\%} \qquad P_{T1} = 3.33\%$$

Step 4 – Updating the past history probability. During this last step, we look back to previously computed stops to update the already computed probability based on the category probability of the current stop. When the probability of the current stop exceeds a given threshold T_2 (that is an application dependent parameter), we can perform a updating of the probabilities of all POIs of all stops already considered for the trajectory T, on the basis of the newly assigned category. In other words, when for a POI the probability is very high (a certain POI) we can use the category criterion to refine probabilities of the POIs already computed. This step is really useful when an uncertain stop happens at the beginning of the trajectory: updating is the only way to increase its level of certainty.

For this step we need a threshold T_2, which represents a threshold for very high probability. For example, we can choose $T_2 = 0.9 + \frac{0.1}{size(\mathrm{L_i})}$. Since T_2 measures the certainty that we have about a POI to be the goal of the stop, T_2 should be closed to 100%.

In this case we can go back to the previous stops of the trajectory and update the probability, when some of the semantic rules fit. Let's consider an example: suppose that at stop *n-1* the algorithm assign a 30% probability to a restaurant, and for the step *n* there is another association to a restaurant greater than T_2. Then the updating step checks the semantic rule set and founds out that two consecutive stops at restaurants are not a common behaviour, unless more than 4 hours are passed. Suppose that the time interval between stop *n-1* and *n* is short, the algorithm updates the probability of the restaurant at step *n-1* to zero and adjusts the others associated to the same stop.

3.2. Semantic Rules and Constraints

Common Sense Constraint Rules

These rules use the specific domain information about the POIs that can be obtained from domain expert, municipality records or geographical services (e.g. Google Maps). These classes of constraints are of two different kinds: CSCa rules and CSCb rules.

CSCa rules exclude some "impossible" or unlikely POIs due to temporal incompatibility. For example, a working place visit typically lasts 6 to 8 hours, therefore we can exclude working POI lasting less than a given threshold, as shown in the following rule.

```
IF(poi_category="WORKING PLACE" AND duration_of_stop<2h)
THEN FALSE;
```

Therefore, combining the category of the POI with time duration of the stop, we can disregard the POI from the list.

CSCa rules, when available, can be added at the end of Step 1 of the AssignPOItoStop algorithm.

CSCb rules represent constraints that use both the <stop, POIs> pair and the POIs already classified in the trajectory. For example:

```
IF(poi_category="EATING PLACE"
       AND previous_stop_category="EATING PLACE")
THEN FALSE;
```

This means that when the current selected POI is an eating place and the previous stop has already been associated to a place for eating, this POI has to be excluded, i.e. it is unlikely that eating is the goal activity of the current stop. The semantic meaning is that a person will not usually go to two consecutively eating places. This heuristics obvious does not capture all the "non standard" cases as people visiting two restaurants during the same evening, for having dinner in a restaurant and a drink in another restaurant. Moreover, the rule can be refined for example by considering the duration of both stops. CSCb-rules can also be procedures that return a category instead of a boolean value and can be added to Step 3 of the *AssignToPOI* algorithm.

Probability Rules

PROa rules are an extension of CSCb rules returning a probability value instead of a boolean value. For example, the following rule asserts the probability that a tourist would go to a museum if she/he has been in a hotel in the morning and it is rainy (these kinds of association can be extracted by statistics or by domain experts).

```
IF(poi_category_stop1="HOTEL"
       AND period_of_the_day_stop1="MORNING" AND weather="rainy")
THEN RETURN poi_category_stopN("MUSEUM", 80%);
```

and this probability value can be used by the algorithm to update the probability of the current POI. It is worth noticing that this kind of rules are pretty sophisticated since we need additional information about the context, such as the weather, the period of the day and general statistics about people habits. Therefore we cannot assume they are always available.

PROb rules are a specialization PROa rules returning a probability function. For example:

```
IF(poi_category="TOURIST place" AND last_N_stop_category="HOTEL"
       AND period_of_the_day="MORNING" AND weather="rainy")
THEN  poi_category("MUSEUM", MIN(10,[80-((N-1)*10)]) %);
```

This rule states that when a tourist has been in a hotel in the morning, there is an high probability that he will go to a museum if it is rainy, but this probability decreases with the increase of the number of stops after the hotel. So the rule says that it is more probable that the tourist will go to a museum immediately after he has been in a hotel.

4. FROM LOCAL TO GLOBAL: TRAJECTORY ACTIVITY

This module assigns a probability to the whole trajectory T on the basis of the probabilities assigned to each POI of each stop of T. For each existing POI category, we compute the probability of the behaviour associated to the trajectory.

In order to assign a behaviour to the whole trajectory, for each activity category defined we compute the global probability. This passage can be done in several ways. The simplest way, that we implemented in the algorithm, is to define a weighed sum for each category. In other words, we sum all the probabilities of all *POIs* belonging to the same category and divide this number by the sum of probabilities.

As an example, consider a trajectory of five stops in which the aggregate probabilities for each category are described in the following table:

STOP 1	STOP 2	STOP 3	STOP 4	STOP 5	TOT
B1=70%	G2=60%	B3=40%	D4=50%	B5=70%	290
G1=30%	B2=30%	G3=39%	B4=40%	G5=20%	159
D1=0%	D2=10%	D3=21%	G4=10%	D5=10%	51

For each category we compute the probabilities:

$$\boldsymbol{B} = \frac{\mathbf{180 + (30 + 40) + 0}}{\mathbf{290 + 159 + 51}} * \mathbf{100} = \mathbf{50.00}\%$$

$$G = \frac{60 + (30 + 39 + 20) + 10}{290 + 159 + 51} * 100 = 31{,}8\%$$

$$D = \frac{50 + (10 + 21 + 10) + 0}{290 + 51 + 159} * 100 = 18.2\%$$

So the current trajectory belongs to the category activity associated with B Place category, with a certainty of 50%.

5. EXPERIMENTS

We run some preliminary experiments to test our approach. The considered POIs refer to the center of the city of Milan (Italy): we classify 39256 POIs in four main categories by grouping them according to a conceptual hierarchy. We assume that each POI belongs to only one main category among:

- SERVICES (4339 POIs), that contains services provided by the city like train stations and metros.
- FOOD (7036 POIs), that contains places related to food is like supermarket and restaurant.
- PERSON ORIENTED (15371 POIs), that contains places related to health (pharmacies), house (furniture) and entertainment.
- ITEM SALE (12510 POIs), that contains places goods are sold, not included in the previous ones.

Moreover, we assign to each POI the average visit time a person spends into the POI, with reference to expert knowledge and some statistics.

The test movement dataset consists of a set of trajectories collected by a GPS device installed on 17000 private cars in the Milano city during a week: each trajectory is composed by a set of stops, each one with a temporal duration. The stops have been computed by the algorithm presented in [10].

We considered only the first day of the monitored week to avoid heavy computations. We first filtered only stops longer than 10 minutes: shorter stops don't characterize a person behaviour since they could be related to measurement errors or traffic congestions. Since the POIs we used in this experiment refer to the center of Milan, we discarded all the stops happening in the suburbs of the city. We also discarded short trajectories with less than 5 stops.

Thus, for the first day, we considered 654 trajectories having from 5 to 15 stops, with an amount of 4527 stops longer than 10 minutes (5.8 stops on average for each trajectory).

The values used for the parameters of our application are:

- the *Velocity* of a walking person is 1.3 m/s (i.e. 4.68 Km/h)
- the *Maximum Distance* that a person accept to walk is 1000 meters.
- Values for thresholds T1 and T2 have been computed according the suggestions of section 3, thus:

$$T_1 = 0.5 + \frac{0.5}{number\ of\ POIs} \qquad T_2 = 0.9 + \frac{0.1}{number\ of\ POIs}$$

The algorithm has been implemented in Java without concurrency: it runs as a Java application on a Windows Vista environment, on a Intel Centrino Duo 2GHZ. It needs only few seconds to run, computing trajectories and updating the database (9 seconds for the 654 trajectories of the first day, 51 seconds for the 5102 trajectories of the whole week). We found that, out of the 654 trajectories, 370 have all their stops associated with only one POI with a percentage of almost 100%, and 249 have all their stops associated at least two POIs. Finally, the accuracy of the classification of trajectories in the four main categories is very high: the 55% of trajectory are classified with a probability rate >80%. In this experiment we did not use any semantic rule. Figure 5 shows a snapshot of the resulting table where each trajectory *id* is associated to a trajectory type (namely the behaviour) and the associated probability value.

	TRAJ_ID	TRAJ_TYPE	TRAJ_TYPE_PROBABILITY
1	24756100	items_sale	0,8268083956868176
2	7305900	items_sale	0,5224954806175307
3	12758800	items_sale	0,7699782336921265
4	30482500	items_sale	0,7391142714611428
5	32902800	items_sale	0,7933920623329205
6	589200	items_sale	0,88870106171125
7	27398700	items_sale	0,8729717936665202
8	22463500	items_sale	0,9424722041360427
9	32834300	items_sale	0,7347639699577474
10	32041800	items_sale	1
11	31663800	person oriented	0,6965128763217244
12	15107900	items_sale	0,9270557313797789
13	7047900	items_sale	0,9258616138097218
14	27177900	items_sale	0,720377772969764
15	32405600	items sale	0.871095886850189

Figure 5 A snapshot of the output

These experiments run so far are preliminary and gave us a first feeling of the behaviour of the algorithm. Of course, more accurate experiments are planned. First of all, we need to compare the results of the algorithm with a “ground truth” to have a measure of the results accuracy.

6. CONCLUSIONS AND FUTURE WORK

In this paper we proposed an approach that computes the most probable visited places of people during their movements and, based on that, infers the behaviour to the whole movement of the tracked person. The algorithm associates a list of probable places of interest (POIs) to each movement stop and, consequently, classifies the overall trajectory into a behavioural class, based on the semantic category of each POI. We also proposed classes of rules encoding domain specific knowledge to refine the POI semantic classification. The approach has been experimented in a real case study on a dataset of trajectories of cars moving in Milan city area.

Ongoing and future works include the enhancement of heuristics to assign probability to stop-POI association, refining the computation with more detailed domain-dependent rules. Another point that can be improved is the management of the threshold. Since the algorithm uses several parameters application dependent, it could be difficult for a user to set the right ones. We intent to exploit the activity ontology to encode in it the application dependant parameters. In this way, the algorithm can use uses these parameters and threshold directly from the ontology.

Of course, we intend to run further experiments to further prove the usefulness of the approach. In this direction we are going to test our algorithm in a different dataset which contains traces of trucks that deliver gas. What is interesting in this dataset is that some of the stops activity are known (e.g. deliver gas, having lunch, etc), and this gave us the ground truth to test our methods.

Another possible research direction to improve our work goes towards the personalization of the method. The parameters of the algorithm may depend on user profiles and this gives the method more effectiveness. However, on large datasets, this may be unfeasible.

Another different direction of research is related to the privacy aspects. Indeed, although the trajectories are anonymized, it has been proved in the literature that the knowledge about the movement of a person may allow to infer the identity of that person and the possible sensitive places she/he has visited. Therefore, privacy-aware analysis methods have to be applied to avoid the disclosing of personal information such as the visit to sensitive places, such as hospitals. Therefore, we need to investigate methods to avoid the disclosure of "sensitive stops", while retaining, as much as possible, the quality of the inferred activity.

7. Acknowledgments

This work has been supported by the EU-FET Coordination Action MODAP.
The dataset has been donated by Octotelematics for research purposes.
We would like to thanks Christine Parent for useful discussions.

References

[1] AIDA, Affective, Intelligent Driving Agent, http://senseable.mit.edu/aida/

[2] Gennady L. Andrienko, Natalia V. Andrienko: Extracting Patterns of Individual Movement Behaviour from a Massive Collection of Tracked Positions. BMI 2007: 1-16

[3] Alvares, L. O., Bogorny, V., Kuijpers, B., Macedo, J., Moelans, B., & Vaisman, A. A model for enriching trajectories with semantic geographical information. Proceedings of the ACM GIS 2007.

[4] Bogorny, V.; Kuijpers, B.; Alvares, L.O. ST-DMQL: A Semantic Trajectory Data Mining Query Language. In: International Journal of Geographical Information Science. Taylor and Francis . pp.1245 – 1276, vol 23 (10)

[5] Baglioni, M., Macedo, J., Renso, C., Trasarti, R., & Wachowicz, M. Towards a semantic interpretation of movement behaviour. Proceedings of 12th AGILE Conference, Lecture Notes in Geoinformation and Cartography, 2009.

[6] Fosca Giannotti, Dino Pedreschi: Mobility, Data Mining and Privacy - Geographic Knowledge Discovery Springer 2008

[7] Peter Kiefer, Klaus Stein: A Framework for Mobile Intention Recognition in Spatially Structured Environments. BMI 2008: 28-41

[8] Spaccapietra, S., Parent, C., Damiani, M.L, Macêdo, J., Porto, F., Vangenot, C (2008) : A conceptual view on trajectories. Data Knowl. Eng. 65(1): 126-146 (2008).

[9] Yu Zheng, Yukun Chen, Xing Xie, and Wei-Ying Ma, Understanding transportation mode based on GPS data for Web applications, in ACM Transaction on the Web, Association for Computing Machinery, Inc., January 2010

[10] Zhixian Yan, Parent Christine, Spaccapietra Stefano, and Dipanjan C. A hybrid model and computing platform for spatio-semantic trajectories, ESWC 2010, Mobility track, 2010