Extracting user habits from Google maps history logs

Christos Sardianos
Dept. of Informatics and Telematics
Harokopio University of Athens, Greece

Multiple Aspect Trajectory Management and Analysis

This work has been developed in the frame of the MASTER project, which has received funding from the *European Union's Horizon 2020 research and innovation programme* under Marie-Slodowska Curie grant agreement No 777695

NPRP Funding
National Priorities Research Program

This paper was made possible by National Priorities Research Program (NPRP) grant No. 10-0130-170288 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Problem definition

- Smart devices' sensors collect time-stamped geographical coordinates of the user.
- We consider two different types of information that can be extracted from user GPS data:
  - Location information: places the user has visited, along with timestamp and duration
  - Trajectory information: the movement of a user from one location to another.
- Using the GPS sensor, we can extract the geographical location and movement of the user.

- Can we identify the user's behavior and habits, based on the analysis of user's location data?
- This can be valuable for many domains such as Recommender Systems, targeted/personalized advertising etc.

Definitions

Stay Point SP	A geographic region, where a user stays over a certain time interval
User trajectory Tr_{ij}	The sequence of GPS points between two stay points SP_i and SP_j
User habits	A routine of behavior that the user repeats regularly and which tends to occur subconsciously

Method overview

To identify user habits based on the analysis of user's location data, we:

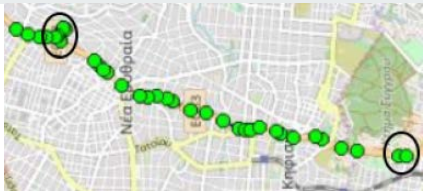
- Analyze user's GPS logs provided through his Google location history.
- Detect user stay points (locations that user usually spends more time).
- Identify the user's frequently preferred transportation types and trajectories.
- Find what type of places the user visits in a regular base (such as cinemas, restaurants etc).
- Extract the habits the user is most likely to have.

Extraction of user stay points

- The extraction of a stay point depends on:
  - a time interval threshold (T_{threh})
  - a distance threshold (D_{threh})
- The set of points that satisfy these thresholds are regarded as the SP.
- The centroid of all GPS points of a stay point is used as the GPS coordinates of stay point SP.
- A maximum radius for the cluster is also kept with the SP.
- Finally, we keep the start and end time-stamp with the stay point.

$$SP = < lat, lon, radius, t_{start}, t_{end} >$$

Extraction of user trajectories



A trajectory Tr_{ij} between two stay points SP_i and SP_j:

- Defines a route comprising a series of GPS locations in chronological order.
- Stay points denote the end of one trajectory and the beginning of another.
- The timestamp difference between two consecutive GPS points in a trajectory exceed the T_{threh}

  they cannot be considered to belong in the same stay point.
- So a user trajectory is characterized by a set of GPS coordinates, a start and an end time-stamp as:

$$Tr = < \{(lat_i, lon_i)\}, tr_{start}, tr_{end} >$$

Semantic enhancement of user stay points

Stay point semantic enrichment process:

- Attaches additional information for characterizing the extracted stay points and trajectories.
- Main sources for this information are POI services with semantic annotations for popular points of interest and common types of POIs (e.g. sport facilities, public buildings etc.).
- Users can add semantic information for their own POIs such as "Home" or "Work".
- After extracting stay points, we access POI information services and search for nearby POIs, at a short range and within the stay point limits.
- As a result, we characterize the type of the stay point, such as: Park, Stadium etc.
- So finally, a stay point is characterized as:

$$SP = < lat, lon, POI_{type}, POI_{category}\ t_{start}, t_{end} >$$

Semantic enhancement of user trajectories

- User trajectory semantic annotation process refers to:
  - Applying data mining techniques to the trajectory points information (latitude, longitude and timestamp).
  - Detecting of user type of movement across the trajectory.
- This allows us to detect whether the user prefers public or private means of transportation etc.
- We treat the detection of user movement type as a classification problem.
- The movement categories can be: motionless, walking, running, riding a bike, driving a car, being on the bus, being on a train/metro, and any other type of movement.
- At the end of this step a trajectory is characterized as:

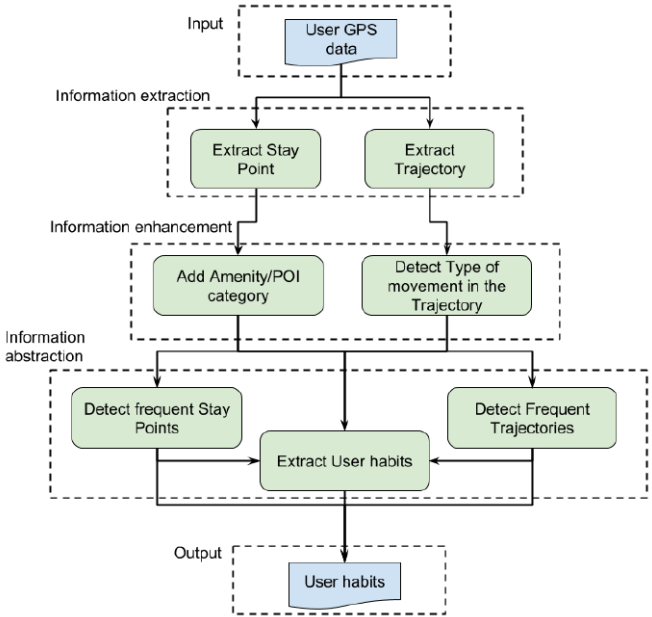
$$Tr = < \{(lat_i, lon_i)\}, POI_{start}, POI_{end}, t_{start}, t_{end} >$$

Abstracting user data to user habits

- In our scenario, habits are repetitive user activities such as being in the same stay point, taking the same trajectory at the same time or day of a week.
- After detecting frequently accessed POIs and trajectories, the abstraction of information at various levels of granularity in time (e.g. time zones, days, etc), POI type, and movement type follows.
- The result of this process comprises frequently occurring staying or movement patterns for one or more users.
- The extracted patterns form the user habits.

Basic architecture of the application

Overall, our approach for analysis can be described as a three-tier analysis with multiple tasks in each tier.



Tier 0: Data collection

- We use data from Google Maps history, which are imported as KML (Keyhole Markup Language) files to the system.
- The same processing pipeline can be applied to the actual GPS data collected by the smartphone instead of using KML files.
- The information extracted from the analysis of a user activity in a certain time-frame can be stored in the phone and all the actual GPS data for this frame can be erased.
- For example, when the user commutes to work, we can store information about the trajectory (e.g. start/end time and location, and probably a few intermediate points) and erase all the intermediate GPS data.

Tier 1: Information extraction

Extraction of user stay points

- We employ DBSCAN clustering to identify the locations where user stays for a certain amount of time.
- To clusters together neighboring (in space and time) GPS traces, the distance of two points is a linear combination of geographic distance and time distance.
- The value for MinPts depends on the frequency of recorded GPS spots, the distance threshold of interest and a moving speed threshold.

Detection of user trajectories

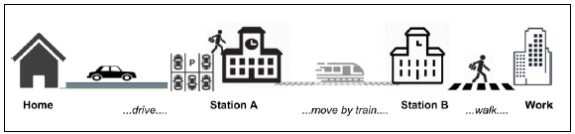
- A large set of stay points may exist in user's GPS logs
- A respectively large set of trajectories are formed among the different stay points.
- The trajectories are directly defined as the sets of GPS tracks between two consecutive stay points.

Tier 2: Information enhancement (user stay points)

- We employ the OpenStreetMaps service, which offers an API for retrieving information about various POIs in a geographical area to characterize stay points.
- For this purpose, a bounding box is created for each stay point, using the GPS coordinates of the stay point as the bounding box center and a range that does not exceed the radius of the respective cluster.
- The OpenStreetMaps API is accessed to retrieve POIs within the geographical area defined by the bounding box and it responds with an XML formatted result.
- From the locations returned, the closest to the stay point is used to characterize the stay point.

Tier 2: Information enhancement (user trajectories)

- For the semantic annotation of a user trajectory, we process all consecutive GPS traces in order to detect user movement speed, user direction and user speed changes.
- We classify each trace individually and then classify the trajectory as a whole.
- A post-processing step aggregates this information for all segments of the trajectory and assigns the movement type that most likely matches to the specific trajectory.



Tier 3: Information abstraction

Frequent user stay points detection

- Analyzing the stay points of a user, we can find if the user tends to visit specific stay points more frequently.
- We cluster stay points using DBSCAN to get clusters with points visited many times by the user.
- We rank clusters in descending order of size and keep the top ranked stay points for a user.
- A first step of abstraction is to find the preferred days or time zones for a user to visit a stay point.
- The result of this step can be similar to the following:

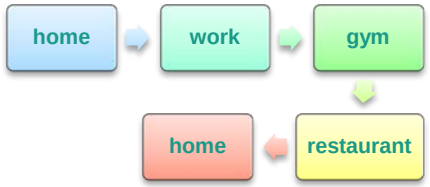
*The user has visited train station A **n** times this month. The preferred days are week days, and the preferred time zones are early in the morning and the afternoon.*

Frequent trajectories detection

- The abstraction of user trajectories will highlight the preferred movement paths and way of movement of the user.
- We considering the set of trajectories that the user has followed to go from one frequent stay point to another.
- We apply the clustering-based sequential mining (CBM) algorithm over the set of trajectories.
- The output of this process is the set of most frequent trajectories followed by the user.

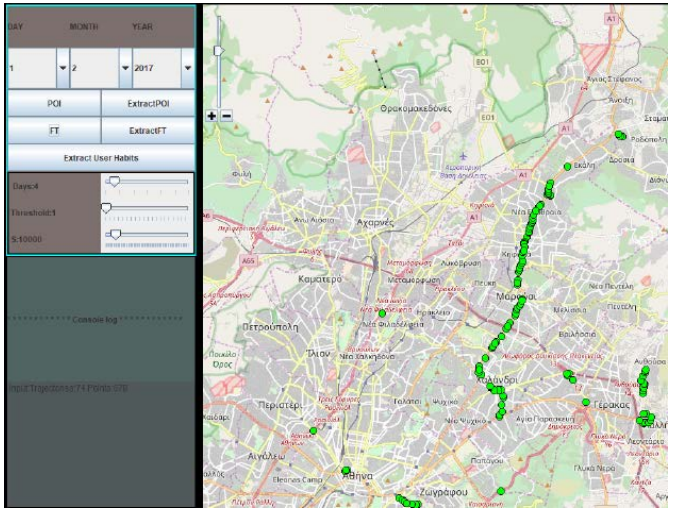
User habits extraction

- We need to find frequently occurring activity patterns in user logs (frequently visited locations and frequent trajectories that the user follows).
- These patterns will define the user's habits.
- We discover user's tendencies on commuting at specific dates and times.
- We treat the user habits extraction as an association rules extraction problem.
- We use different levels of abstraction:
  - Different places that have been visited by the user and have been annotated as restaurants, bars or cafeterias can be generalized to the category amenity and lead to rules with stronger support
  - The time-stamp information can be mapped to day zones (e.g. morning, afternoon, evening) or days
- The output of this information abstraction step is fed to the Apriori algorithm to extract user's association rules.



Demo application

- The application is written in Java and is available as a standalone Java program.
- We use user's Google Maps History files (KML files) as input and processes them.
- Using these files as input, we can extract the user's frequent stay points and trajectories, which can be displayed over an OpenStreetMap layer or exported to separate files.
- The analysis of user stay points leads to a set of user's tendencies.




```
Latitude        Longitude       Time
37.8883883      23.9400781      Sun Jan 01 18:11:51 EET 2017 - Sun Jan 01 22:58:04 EET 2017

Location frequence:45
Frequent Day: Sunday

Latitude        Longitude       Time
37.9033496      23.7499417      Sun Jan 08 20:00:25 EET 2017 - Sun Jan 08 22:51:04 EET 2017

Location frequence:40
Frequent Day: Sunday
Location frequence:37
Frequent Day: Monday

Place name:Barón
Amenity:cafe    ID:3942492230
```

Demo application

- These files are fed to the Apriori algorithm to extract user habits.
- The input data for the Apriori consist of:
  - the user's extracted stay points combined with
  - the time-stamp of occurrence in daytime zone (e.g. morning, afternoon, night) and day (e.g. weekday, weekend) format
  - the type of movement and/or
  - the type of the amenity
- So, when the Apriori algorithm is fed with information in the form of:

  {DayZone, DayType, MoveType, Dest_POIType}

- The extracted user habits are similar to:

1. **MoveType = Metro ⇒ IsWorkingDay = true**
2. **DayZone = Evening Tag = public_transport 331 ⇒ IsWorkingDay = true**
3. **DayZone = Evening MoveType = Metro Tag = public_transport ⇒ IsWorkingDay = true**
4. **DayZone = Evening MoveType = Metro ⇒ IsWorkingDay = true**
5. **Tag = shop ⇒ IsWorkingDay = true**

What is next?

- We process user GPS logs to extract rules and patterns that describe user's habits.
- As far as the results show, we can identify interesting patterns for the user daily activity.
- Since the parameter selection of the algorithms has been made after experimentation, we consider that a more thorough evaluation of different parameter settings must be examined.
- It is in our intention to adapt our work so far in a Recommender System for real-time and real-life recommendations exploiting user habits.

Thank you!