Mining Trajectory Data

Douglas Alves Peixoto
u5312727

November 1, 2013

Abstract. *Analysis of geographic data has become an important field of study in computer science due to the increasing number of data generated by electronic devices capable to collect geographic information from individuals, such as smart phones and GPS devices. This massive quantity of data, generated by such devices, has led to a rise in the number of research projects and techniques aiming to analyse and retrieve useful information from these sizeable datasets. The goal of this project is to explore GPS trajectories from different individuals and both study and apply computational techniques to retrieve useful information from those trajectories, such as locations of interest and individuals similarity, and later build a tool for data visualisation. This paper shows how geographic data can be processed to retrieve useful information using data mining techniques and how such information can be useful to understand individuals and locations in a region.*

1. Introduction

Large-scale geo-data analysis has become an important field of study due to the increasing volume of geo-information obtained from mobiles and GPS devices. One example of an application of geo-data analysis is to understand individuals' behaviour and movement patterns in cities.

However, the massive amount of data to be analysed has lead researchers to develop computational tools and data mining techniques combined with machine learning algorithms to enable a better management and understanding of geographic information [4]. The importance of the data mining studies for the modern society is to help individuals and companies to extract useful information from large data sets [5][11], which is manually impracticable; hence, these techniques can be likewise applied to analyse spatial data in a large scale.

By learning urban trajectories, for instance, it may be possible to people to make better decisions when visiting an unfamiliar location and to find interesting places in cities [15][1], and to know how people interact with each other. The significance and contribution of this project lies on how to efficiently analyse massive amounts of spatial data and retrieve useful information from it. The goal is to analyse individuals' GPS trajectory data and retrieve features based on their geographic location over the time; such features include Stay Points (SP) and Points of Interest (POI) which can be useful to understand users' interaction and similarity, and both understand individuals' movement patterns and find interesting places in a certain location.

This paper will describe a simple approach to find Points of Interest in two steps, firstly detecting individuals' Stay Points, which are regions where and individual spent a considerable time in a single trajectory; and secondly clustering Stay Points using a density-based clustering algorithm to finally extract POIs. This paper will later describe how to calculate similarity between users using the Points of Interest previously mentioned, and rank both users and Points of Interest using a HITS-based approach.

Geographic information (e.g. travelogues, GPS data, image tags) can be collected from different sources such as smart phones, GPS devices, social networks, etc.; many of these data can be integrated into a unique dataset even though they may have been generated from different devices. For the purpose of this paper it was used GPS data provided by the GeoLife project, Zheng et. al [15][16][17], which is a social network service that incorporates users, locations and GPS trajectories. All the GPS data used in this project were collected from the GeoLife dataset. Additionally, this paper will present a simple Web tool developed for the purpose of data visualisation, allowing a better understanding of the achievements.

The remainder of the paper is organised as follows. In Section 2 we discuss the related works. In the Section 3 is described the methodology and a review of the GeoLife project. Finally, the results of this paper as well as the tool for data visualisation are presented in Section 4 while a conclusion is drawn in Section 5.

2. Related Works

This study will draw on a range of approaches to the mining of users' trajectories and locations, by using the GPS data provided by the GeoLife network service. Several research efforts have been dedicated to this problem. Lu et. al [8] present a novel framework to automatically help users to plan a trip to an unfamiliar location by using tagged images (e.g., Flickr Images) and textual travelogues. By mining these artefacts, and based on previous tourists experiences and user's personal preferences such as the season, travel duration and visiting destination, they might predict representative locations, typical stay times, and

popular travelling routes within the given user's destination. A similar approach is used by Choudhury et. al [1] for automatically construct travel itineraries using Flickr tagged images.

Similarly, Hao et. al [6] propose a framework to generate a representative and comprehensive location overview by mining textual travelogues. Given a set of travelogues collected from the Web shared by previous travellers, it is possible to extract information and generate location-representative tags (e.g., Central Park, Statue of Liberty, Times Square). These representative tags can be combined with the location name (e.g., New York) to retrieve images related to a location from the Web, providing to the user a visual and textual description of a given destination.

To support travellers to plan a trip itinerary Zheng et. al [15][17] propose the social networking service GeoLife to understand users, locations and trajectories in a collaborative manner based on users' GPS logs. The goal is to manage GPS trajectories using machine learning algorithms to build a trip recommender by mining interesting destinations. This analyses users' activities with the purpose of predicting user's interest in a given location, finding experienced users in a certain geo-region and searching trajectories by location. The same GPS trajectories data set, which is also the data set this paper is interested in, was used by Zheng et. al [16] to predict transportation modes between locations in a geo-region, supporting users to know not only which are the typical locations of interest in a given region, but also how such locations can be reached.

This paper will basically focus on the task of retrieve individuals' points of interest and find users similarities based on their stay points; for this purpose this project will mainly focus on the works proposed by Zheng et. al [15][16][17] and others similar approaches as those mentioned above.

3. Methodology

This project will look toward to an analysis of GPS trajectories; this section outlines the methodology adopted to achieve the goals of this project as described in section 1.

Figure 1 illustrates seven steps of the methodology applied in this project, which will be further described in this section. Firstly we introduce the GeoLife GPS dataset (1), which will be used in our GPS data analysis. The second step correspond to the data cleaning technique (2), where will be introduced a method based on coordinates speed to eliminate noisy data. The Step 3 of our geographic analysis regards to the detection of Stay Points (3) based on the time an individual stayed within a region. Steps 4 and 5 illustrate how Stay Points are clustered (4) to further detection of Points of Interest (5). Finally, in steps 6 and 7 we demonstrate how the previously calculated POIs can be used to calculate similarity among users (6) based on the number of times each pair of individual has visited the same location, and ranking users and locations (7) by means of a HITS-based approach.

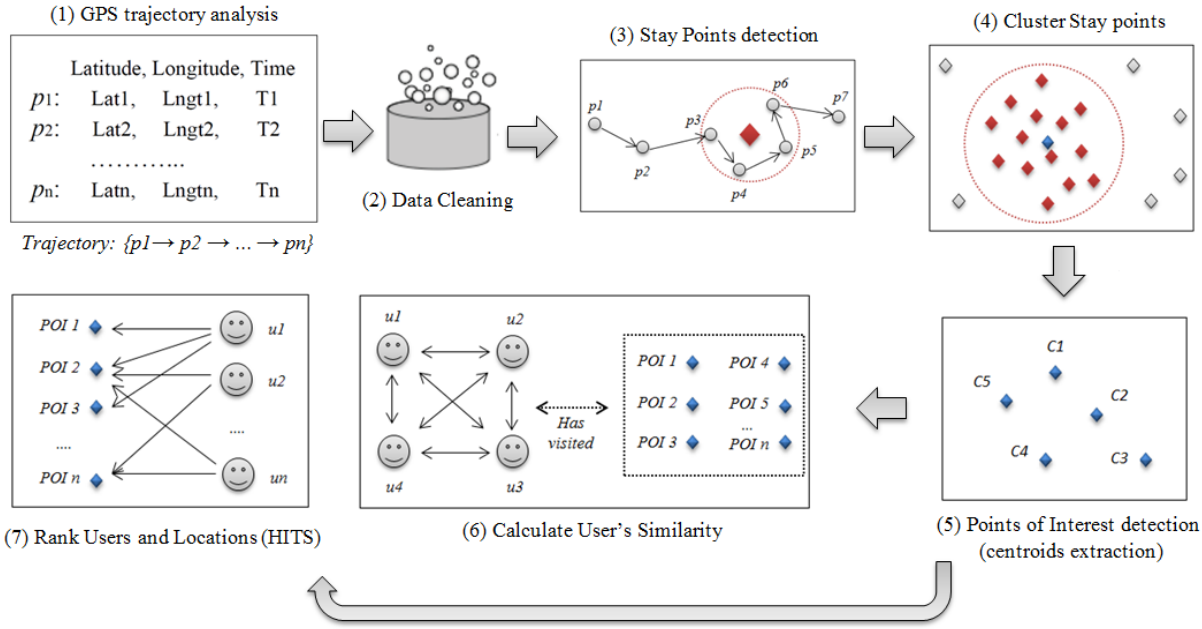


Figure 1. Project methodology and achievements

3.1. Geolife overview

In this section, we briefly introduce the GeoLife networking service Zheng et. al [15][16][17]. As previously mentioned, this paper is mainly based on the GeoLife project. The GeoLife dataset includes individuals movements in life routines, such as go to work or go to home, and also outdoor and entertainment activities. Therefore, due to its variety and flexibility of data, all GPS trajectories used in this project were collected from this dataset.

GeoLife Dataset	
Number of Users	182
Number of Trajectories	18.670
Number of Coordinates	24.876.978
Total Distance	1.292.951 km
Total Duration	50.176 hours
Effective Days	11.129

Source – GeoLife user guide [9]

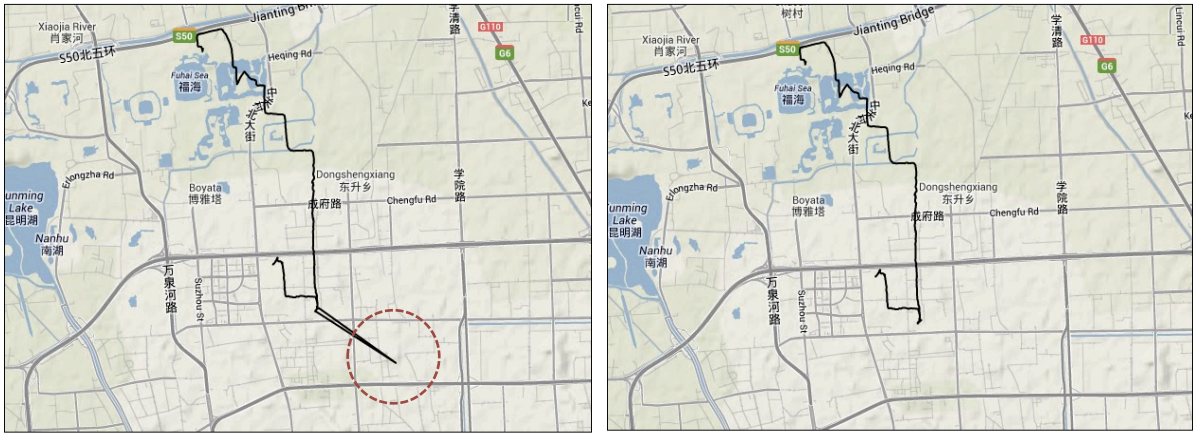
Table 1: GeoLife dataset version 1.3

The GeoLife dataset contains GPS trajectories collected by (Microsoft Research Asia) among 182 individuals in a collaborative manner [9]. Information about this dataset is shown in Table 1. The data was collected in a period over three years and were recorded by different devices such as GPS loggers and GPS-phones. The trajectories in this dataset are composed by a sequence of raw GPS coordinates with time-stamp, latitude and longitude. The majority of these trajectories were recorded in a dense time and distance interval, for instance every 1~5 seconds or every 5~10 meters per coordinate.

3.2. Data Cleaning

The first step in the analysis and processing of the dataset is to remove possible noise from the data. Data cleaning is a data mining technique to detect and either remove or correct inconsistencies or missing data in a dataset Han et. al [5], such inconsistent data may affect the results of the study. Noise in the data may be caused by many different reasons, such as error in electronic devices (e.g. GPS loggers), software error or human mistake. After cleaning the data must be consistent with the other similar data in the system.

Firstly we must detect noise into the dataset, for this purpose we first chose to make a visual checking to initially find any possible irregularity in the trajectories. Figure 2 (a), for instance, shows that there is a single point on the trajectory, rounded in red, that deviates from the trajectory pattern; at this point an individual suddenly took a very high speed, more than 200 km/s, in period of less than 5 seconds, what is quite improbable. To remove this type of noise a solution based on the individual speed along the trajectory was adopted; we first calculate the speed an individual took from every point on the trajectories and then check if in between any two points the individual's speed fluctuates to high values.



a) Before cleaning b) After cleaning

Figure 2. GeoLife trajectory data cleaning

Figure 3 summarises over 100.000 values for speed of individuals taken between trajectory points. Notice that there are some single points fluctuating to a high speed in a very short period of time if compared with its previous point. Hence, the threshold of 100m/s was chosen and all single points over this mark have been removed. This strategy was applied for the whole database. Figure 2 (b) shows the result of the cleaning process over (a).

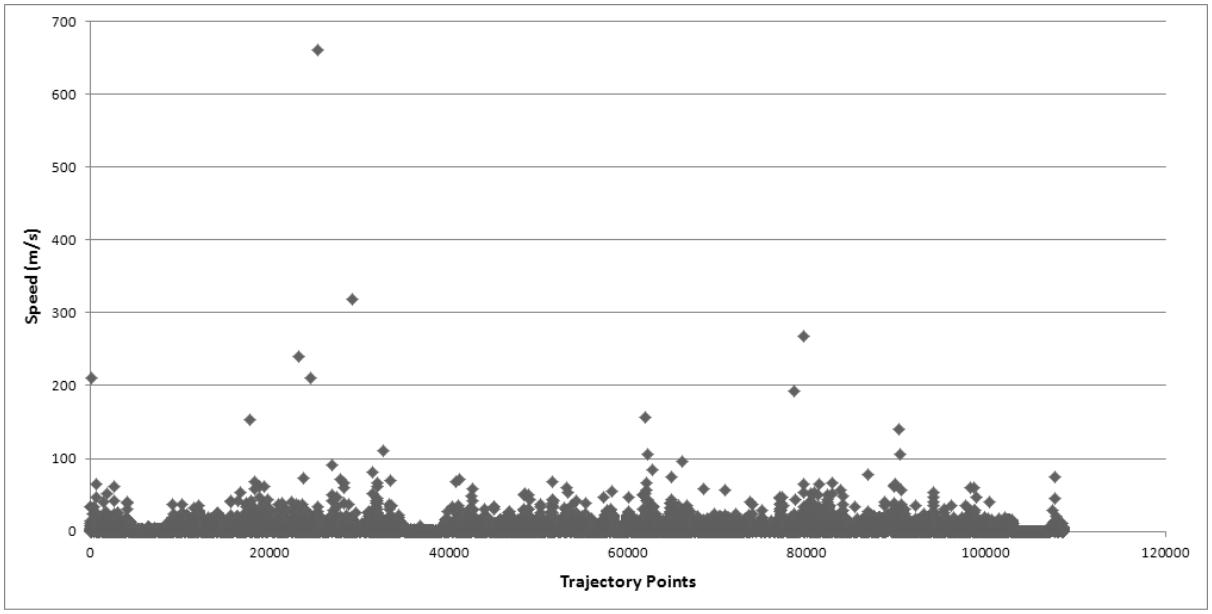


Figure 3. Individuals' speed between trajectory points

3.3. Stay Points detection

After the data cleaning we can take the first step toward to the computation of users' Stay Points, and after that calculate Points of Interest, which is one of the goals of this paper. Points of Interest will be defined further in section 3.4.

Stay Points are geographic regions were an individual has spent a considerable time on its surroundings. The first step to compute Stay Point is to cluster individuals' points based on its time and distance taken on a trajectory into a geographic region. For this purpose, a simple time and distance based clustering algorithm was developed in this project. Remember that a trajectory is a set of *n* consecutive points *{p1, p2, ..., pn}* each one with latitude, longitude and a time interval.

In Algorithm 1 a coordinate *i* of a trajectory (path) is chosen as a candidate point, then the distance between each coordinate *j* after *i* in the trajectory is calculated. If the distance between coordinates *i* and *j* is below to the *DistanceThreshold* the coordinate *j* is add to the cluster, otherwise the cluster is finished here because it means that the individual has left the

range of our distance threshold. Because we are dealing with spherical geometry, the *Distance* function here is the *Haversine Distance,* for any two points on a sphere the *Haversine Distance* is given by:

$$d = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\emptyset i - \emptyset j}{2}\right) + \cos(\emptyset i)\cos(\emptyset j)\sin^2\left(\frac{\varphi i - \varphi j}{2}\right)}\right)$$

where r is the *Earth* radius, $\emptyset$ and φ are respectively the latitudes and longitudes of points i and j.

Algorithm 1: ClusterPathPoints(PathList, DistanceThreshold, TimeThreshold)

```
1   ClustersSet = {}
2   FOR EACH Path IN PathList DO
3         FOR i FROM 1 TO Path.CoordinateList.size DO
4               Coordinate_i = Path.CoordinateList.get(i)
5               // New cluster
6               Cluster = {}
7               Cluster = Cluster ∪ Coordinate_i;
8               OutOfRange = false
9               FOR j FROM i+1 TO Path.CoordinateList.size AND !OutOfRange DO
10                    Coordinate_j = Path.CoordinateList.get(j)
11                    // Validade by distance Threshold
12                    Distance = Distance(Coordinate_i, Coordinate_j)
13                    IF Distance <= DistanceThreshold
14                          Cluster = Cluster ∪ Coordinate_j
15                    ELSE
16                          OutOfRange = true
17                    END IF
18              END FOR
19              // Get the time spent within the cluster
20              Time = Cluster.LastPoint.time - Cluster.FirstPoint.time
21              // Validade by the time threshold
22              IF Time >= TimeThreshold
23                    ClusterSet = ClusterSet ∪ Cluster
24                    // Go to the next non-clustered point
25                    i = i + Cluster.numberOfPoints
26              ELSE
27                    i = i + 1
28              END IF
29        END FOR
30  END FOR
31  RETURN ClustersSet
```

Next the algorithm verifies whether the time an individual spent into the region (cluster) is longer than the *TimeThreshold* to finally add it to the *ClustersSet*. Notice that for a trajectory $p1 \rightarrow p2 \rightarrow ... \rightarrow pn$ we have time intervals $t1 < t2 < ... < tn$. The algorithm then repeats the previous steps for all trajectories into the dataset. The asymptotic complexity of Algorithm 1 is $O(n^2)$, where n is the total number of coordinates into the dataset.

In this experiment, was set *DistanceThreshold = 200m* and *TimeThreshold = 20min,* in another words, if an individual stay over 20min within a region of 200m, a cluster is detected. Hence, each cluster represents a stay region for an individual.

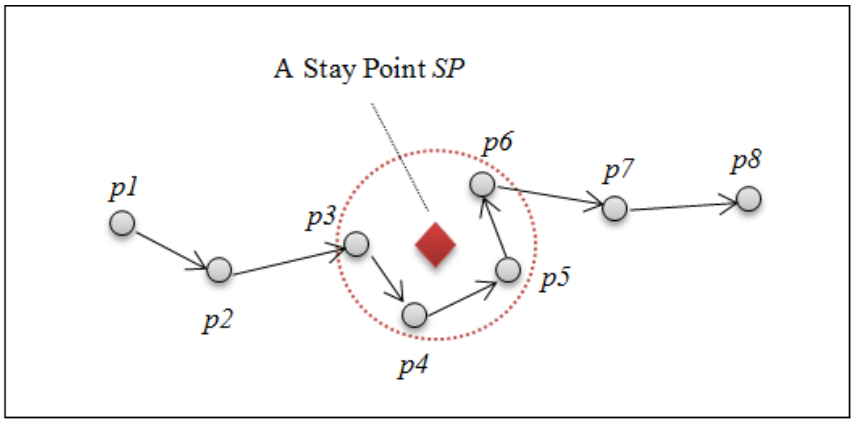


Figure 4. Stay Points detection

Using the clusters computed in the previous phase, we can finally calculate Stay Points. Here a Stay Point is a high level representation of a cluster of points. Thus, it may be represented by the cluster's centroid. A centroid is simply the mean of the coordinates (latitude and longitude) of the points within the cluster. Algorithm 2 shows how Stay Points are extracted from each cluster.

Algorithm 2: getStayPoints(PathList, DistanceThreshold, TimeThreshold)

```
1  StayPointsSet = {}
2  ClustersSet = ClusterPathPoints(PathList,DistanceThreshold,TimeThreshold)
3  FOR EACH Cluster IN ClustersSet DO
4         // Extract centroids from clusters
5         StayPoint = centroid(Cluster)
6         StayPointsSet = StayPointsSet U StayPoint
7  END FOR
8  RETURN StayPointsSet
```

Figure 4 exemplifies the process of Stay Points detection for Algorithm 1 and Algorithm 2. The importance of detecting Stay Points in this phase is to further calculate Points of Interest.

3.4. Points of Interest

In this section will be given a definition of Points of Interest. Having calculated the Stay Points in the previous phase, we are now able to detect locations where individuals have frequently spent a considerable time on its surroundings. To find such a places, we run the two following steps:

1) Firstly we cluster Stay Points using a density-based clustering algorithm [5][11] to find clusters with at least *k* points within it. These new clusters represent frequently visited regions, at least *k* times, thus, they are highly likely to be regions of interest.
2) Secondly, similarly to Algorithm 2, each cluster is represented by its centroid (gravity centre), which are the so-called Points of Interest. These locations can be, for example, a restaurant, a shopping centre, a university building or a touristic attraction.

For this purpose, in this work was chosen the density-based clustering algorithm DBSCAN Ester et. al [2]. DBSCAN is a well-known clustering algorithm and works well with large geographic dataset and likewise can be adapted for any distance function, we use *Haversine Distance* here as well. DBSCAN can also detect noise, what in this case is the set of Stay Points that do not belong to any cluster. Figure 5 exemplifies the main idea of density-based clustering with noise detection.

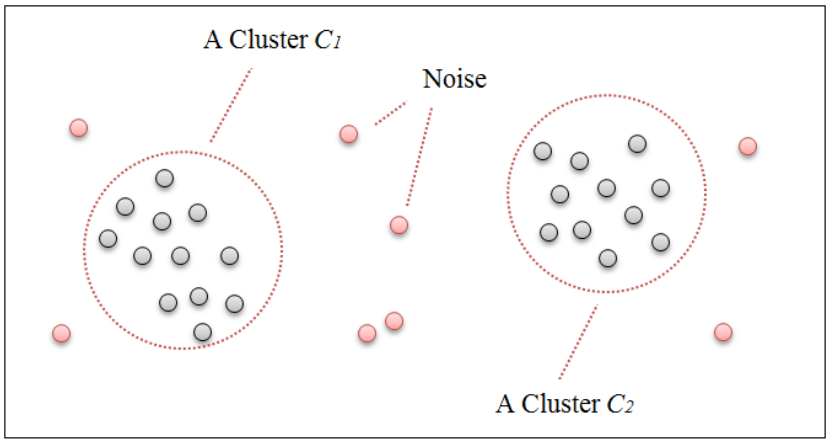


Figure 5. Density-based clustering

DBSCAN requires two parameters: The ε distance threshold (*Eps*) and the minimum number of points within a cluster (*MinPts*). To estimate these two parameter, Ester et. al [2] proposed a heuristic to determine them with regards to the "thinnest" cluster in the database.

The *k-dist* heuristic consist in calculate the distance between each point in the dataset to their k-nearest neighbours. Then create a sorted graph with the result in descending order. The *Eps* threshold is the value in the first valley of this graph. In their experiment, the authors indicate that for $k > 4$ the graph does not significantly change if compared with a *4-dist* graph, thus, in this experiment we set $k = MinPoint = 4$.

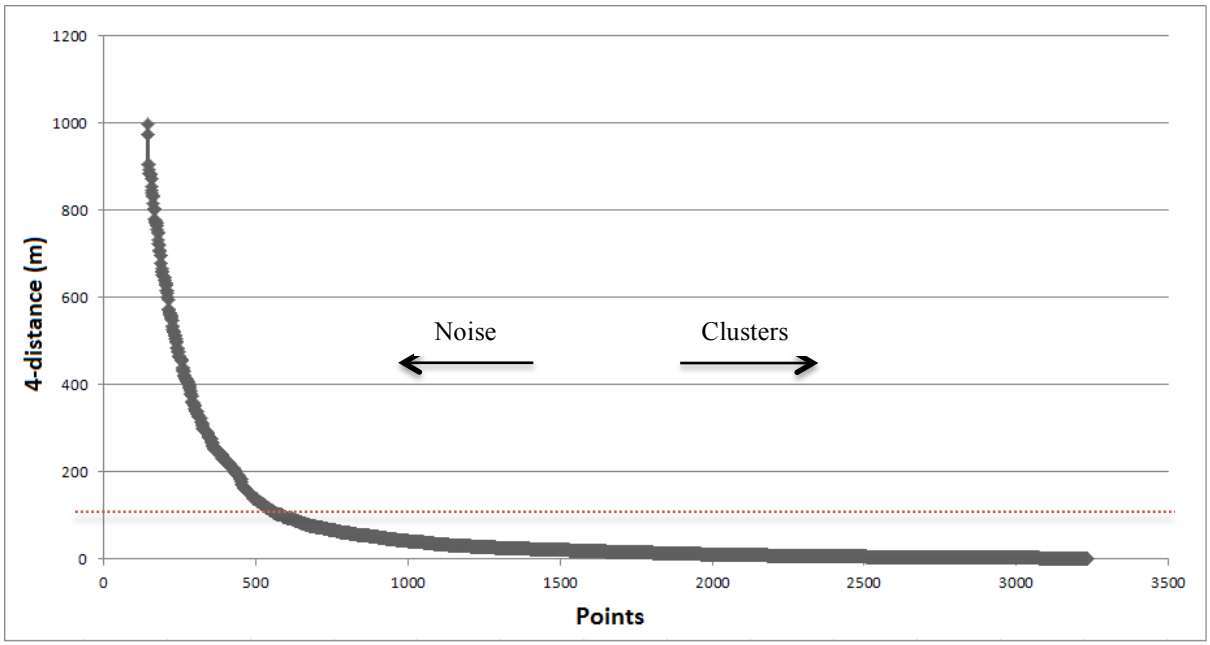


Figure 6. 4-dist graph

Figure 6 shows the *4-dist graph* for our Stay Points dataset, values higher than *1000m* have been omitted for scaling reasons. After an analysis of our *4-dist graph* we chose $\varepsilon = 100m$ as the *Eps* threshold.

For this experiment we have ran the DSCAN algorithm in two different ways. First we cluster Stay Points by individual, thus finding the Points of Interest of a singular individual. Second we run the algorithm for all Stay Points dataset, finding those Points of Interest visited by many individuals. The results for the application of this method over the Stay Points will be given in section 4.

3.5. Users Similarity

Given the previously calculated POIs, it is now possible to calculate similarity among users by simply checking the number of times each pair of individual has visited the same region (POI).

For this purpose, here we use the *Jaccard Similarity* between users, firstly calculating the intersection set and union set of regions visited for each pair of users. Be *A* the set of POIs visited by a user *u1* and *B* the set of POIs visited by a user *u2,* the intersection set *(A ∩ B)* and the union set *(A ∪ B),* for both users, are respectively the set of POIs that both individuals have visited and the set of regions visited for at least one of them. The *Jaccard Similarity* between *u1* and *u2*, denoted here as *Sim(u1, u2),* is simply the product of their intersection set size over their union set size:

$$Sim(u1, u2) = \frac{|A \cap B|}{|A \cup B|}$$

Figure 7 illustrates the set of POIs visited by two users *u1* and *u2*. For this situation, the *Jaccard Similarity* between these two users is given by:

$$|A \cup B| = |POI\{1, 2, 3, 4, 5, 6, 7, 8\}| = 8; \qquad |A \cap B| = |POI\{1, 2, 7\}| = 3$$

$$Sim(u1, u2) = \frac{3}{8}$$

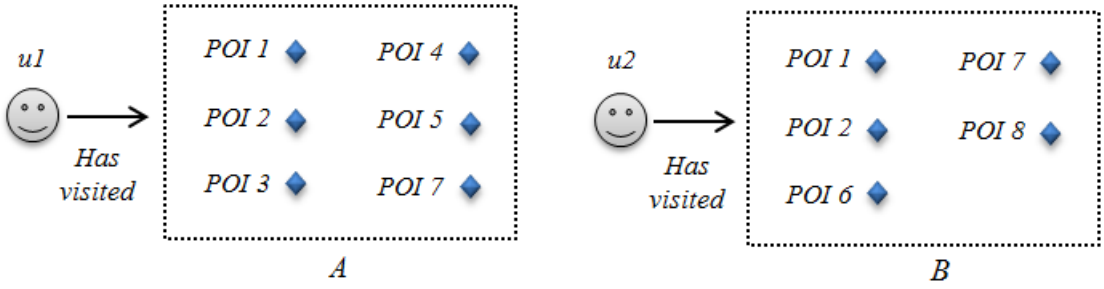


Figure 7. Users set of POIs visited

Groups of people with relatively high similarity are more likely to be potential friends and have similar location preferences, such as bars, restaurants, tourist attractions, shops, etc. This can be useful to create a social network of users with the same interests (e.g. travel locations). In section 4 will be presented a result table for the application of this the method over the users and POIs.

3.6. Ranking Users and Locations

Ranking is another useful approach to understand both users and locations. By ranking individuals, for instance, it is possible to find those with a good knowledge of a region; a location rank, on the other hand, can be useful to find locations of interest within a region.

In this experiment we present a HITS-based model to infer individuals experience and locations of interest from our dataset. HITS was initially build for Web information retrieval [7], its main concept is to score Hubs and Authorities. Figure 8 shows the basic idea of HITS model. In terms of Web page ranking, a hub is a page with many out-links and an authority is a page with many in-links. The hub score is the sum of the scores of the authorities this hub points to, and the authority score in turn is the sum of the scores of the hubs that links to that page. Thus, both hubs and authorities have a mutual reinforcement relationship.

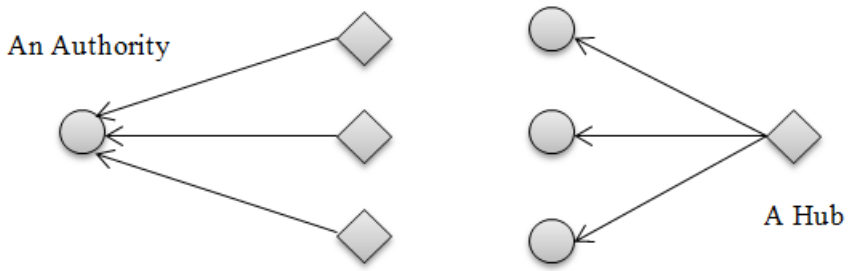


Figure 8. The basic concept of HITS model

Similarly to the Web page ranking problem, in this work a hub is an individual who has visited many places, and an authority is a location (POI) which has been visited by many individuals. Zheng [15] proposed a similar HITS-based inference model to infer users' travel experiences and interest of a location in a given region.

Algorithm 3 shows the pseudo code of an iteration method for HITS used in this work. Here we initially set all hubs and authorities with score equals 1, after that two updates are made, firstly for the authorities score and then for the hubs score. The final scores for authorities and hubs are determinate after k repeated iterations of the update steps. Directly apply the iteration process will lead to diverging score values, to avoid that, we normalize the scores for both hubs and authorities after every update step, thus this values will eventually converge [13]. In section 4 will be presented a result for the application of Algorithm 3 in our dataset.

The ranking process can be useful to generic travel recommendation, if applied for GPS travelogues for instance, providing an user with the top interesting location in a region based on the locations' score [15][17], thereby an individual can easily understand an unfamiliar region.

Algorithm 3: HITS()

```
1  HubList = List of Users
2  AuthorityList = List of POIs
3  // Initialise scores = 1
4  FOR EACH Hub IN HubList DO
5         Hub.score = 1
6  END FOR
7  FOR EACH Authority IN AuthorityList DO
8         Authority.score = 1
9  END FOR
10 // Run the algorithm for K steps
11 FOR Step FROM 1 TO K DO
12        Norm = 0
13        // Update authorities score first
14        FOR EACH Authority IN AuthorityList DO
15               Authority.score = 0;
16               // incomingHubList is the list of Hubs that links to Authority
17               FOR EACH Hub IN Authority.incomingHubList
18                      Authority.score += Hub.score
19               END FOR
20               // Calculate the sum of the squared scores to normalise
21               Norm = Norm + sqrt(Authority.score)
22        END FOR
23        Norm = sqrt(Norm)
24        // Normalise the Authorities score
25        FOR EACH Authority IN AuthorityList DO
26               Authority.score = Authority.score / Norm
27        END FOR
28        Norm = 0
29        // Now update hubs score
30        FOR EACH Hub IN HubList DO
31               Hub.score = 0;
32               // outgoingAuthorityList is the list of Authorities
33               // that Hub links to
34               FOR EACH Authority IN Hub.outgoingAuthorityList
35                      Hub.score += Authority.score
36               END FOR
37               // Calculate the sum of the squared scores to normalise
38               Norm = Norm + sqrt(Hub.score)
39        END FOR
40        Norm = sqrt(Norm)
41        // Normalise the Hubs score
42        FOR EACH Hub IN HubList DO
43               Hub.score = Hub.score / Norm
44        END FOR
45 END FOR
46 RETURN {HubList, AuthorityList}
```

3.7. Data Visualisation

Data visualization aims to clearly represent data in a graphical manner [5], serving as an important technique in data mining and data analysis, in special for the case of geographic information, where basically all the data can be plotted into a map. Data visualization can be efficiently used to discover data relationships that may not be easily observable by looking at the raw data.

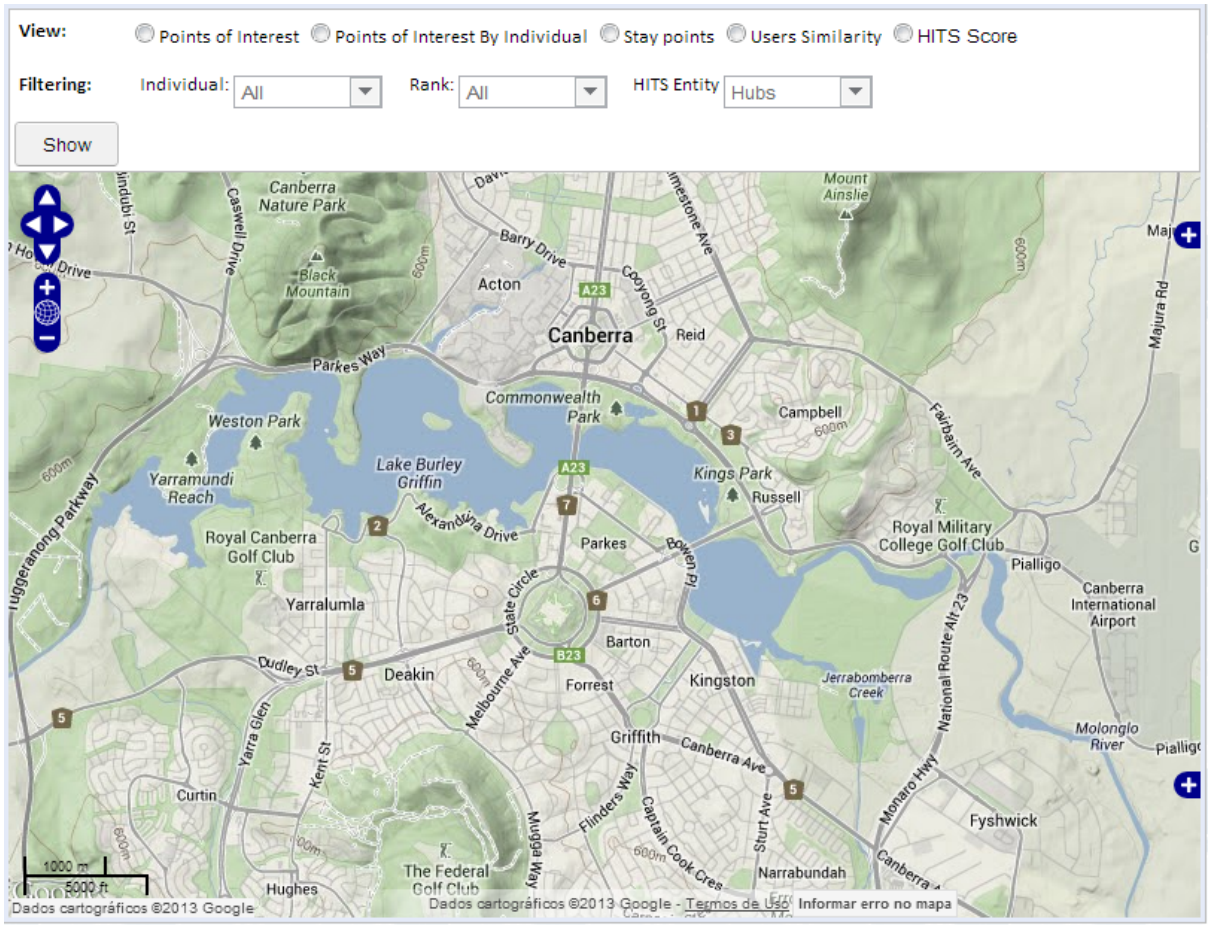


Figure 9. Data visualization tool prototype

In this work was developed a simple visualization tool based on Web technology integrated with an API for map visualization. A prototype of this tool is shown in Figure 9. This tool was developed using the Google Web Toolkit (GWT) technology due to its facility to create and maintain complex front-end web applications. For the purpose of map visualization was chosen the OpenLayers API [10], which is an open source library for displaying dynamic maps on web pages. Another advantage of this API is that it can be easily integrated with GWT.

The results of this project will be discussed in the next section, and some examples regarding the usage of this tool will be given, such as displaying Points of Interest on the map and users and locations ranking using our HITS-based model.

4. Results

For the simplicity of this experiment, GPS trajectories for 10 individuals have been chosen from the entire GeoLife dataset for the evaluation of our methodology. Information about the data used in this experiment is given in Table 2.

Dataset Usage	
Number of Users	10
Number of Trajectories	1.408
Number of Coordinates (before data cleaning)	1.905.811
Number of Coordinates (after data cleaning)	1.905.581

Table 2: GeoLife dataset usage in this experiment

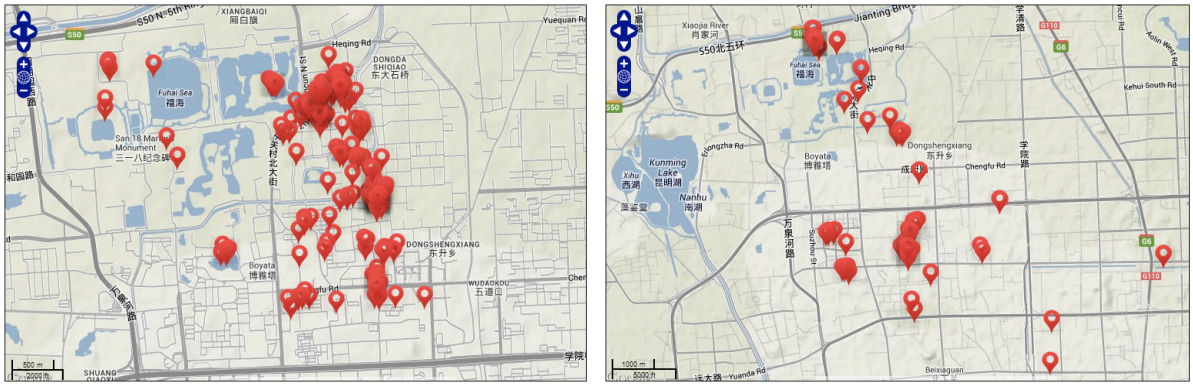
Table 2 also presents information about the number of coordinates before and after running our data cleaning process as described in 3.2. A total of 230 noise points have been removed from the original dataset. All results presented in this works will be performed with the data provided above.

4.1. Results related to Stay Points

Table 3 shows the number of Stay Points calculated for each individual after running Algorithm 2, simultaneously it presents the number of coordinates selected by our clustering algorithm. The results for two individual plotted in our visualization tool in the city of Beijing in China are presented in Figure 10. Remember that, as mentioned in 3.3., the thresholds for Stay Points detection are *200m* in *20 minutes*.

Individual	Number of Paths	Number of Coordinates	Coordinates Selected	Stay Points Detected
1	171	173.856	58.147	386
2	71	108.594	29.544	117
3	175	248.150	73.219	275
4	322	485.182	180.762	903
5	395	439.371	118.163	1.070
6	86	109.010	57.367	159
7	28	31.830	6.651	31
8	34	77.903	17.245	66
9	49	84.599	49.269	121
10	77	147.086	19.409	106
Total	1.408	1.905.581	609.776	3.234

Table 3: Stay Points detection table



a) Individual A b) Individual B

Figure 10. Stay Points coordinates

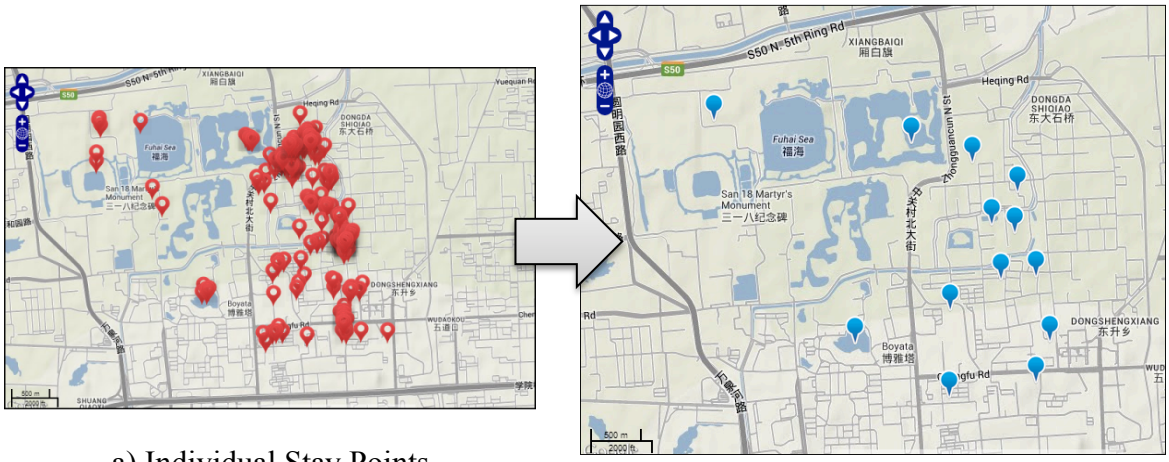
4.2. Results related to Points of Interest

One of the goals of this project is to show how users' GPS trajectories can be managed to detect Points of Interest in a given region. Table 3 provides the results for POIs detection after running the DBSCAN algorithm over the Stay Points by each individual, with *Eps = 100m* and *MinPoints = 4*.

Individual	Number of Stay Points	Stay Points Selected	Points of Interest
1	386	286	14
2	117	65	6
3	275	202	9
4	903	705	22
5	1.070	867	22
6	159	132	8
7	31	10	1
8	66	23	4
9	121	110	4
10	106	63	2
Total	3.234	2.463	92

Table 4: Points of Interest detection table by individual

For the results shown in Table 4, the clustering process was performed for each individual's Stay Points separately, in order to detect those locations which are of interest for every particular user. Figure 11 shows POIs detected for a particular individual (b) from its respective Stay Points (a). Notice that if we compare these points in (b) with those in (a), they precisely correspond to the centroids of regions with the most density of Stay Points.

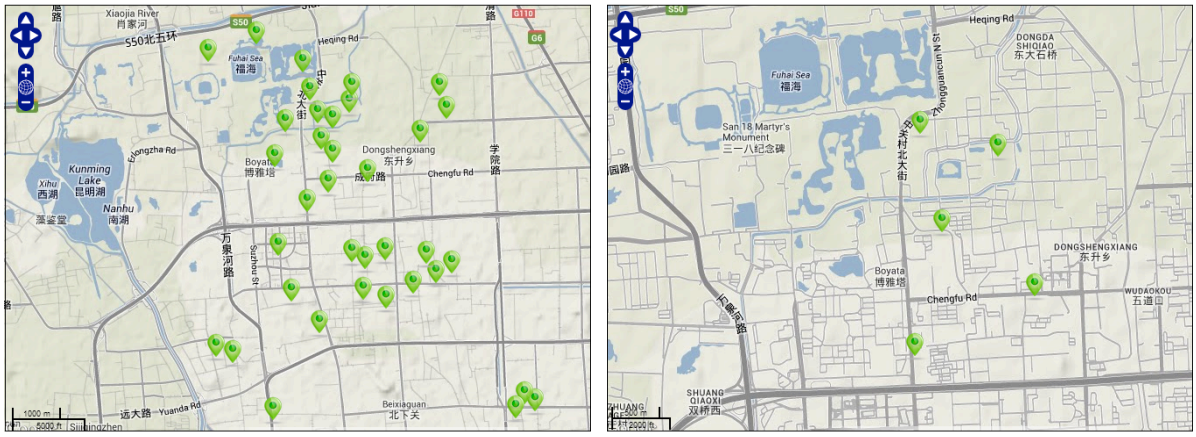


a) Individual Stay Points

b) Individual Points of Interest

Figure 11. Points of Interest by individual

In addition to individuals' POIs, the clustering process was still performed for all Stay Points dataset, finding those Points of Interest visited by many individuals. For the previously given thresholds, the DBSCAN algorithm placed 2.700 Stay Points into 76 clusters. Figure 12 (a) shows those POIs detected after performing the clustering for all the Stay Points dataset in the city of Beijing. In Figure 12 (b) are presented the top 5 POIs in the same region. The ranking criteria provided here by our visualization tool take into account only the number of times individuals have been within each POI region, it differs from the HITS method results, which will be shown later.



a) POIs over Beijing b) Top 5 POIs

Figure 12. Points of Interest

Points of interest shall correspond to different types of location depending on the GPS source. Since GeoLife is a dataset mainly composed by GPS logs of academic individuals, the POIs here can be a university building or either a residence or a restaurant in a university campus as well as a research centre. Indeed, for the city of Beijing, the majority of the POIs are building in Beijing University, Tsinghua University and Microsoft Research Asia as well as some leisure places such as the Yuanmingyaun Park. Furthermore, this approach can be applied for any source of GPS records. For instance, if applied for GPS travelogues, the POIs can represent touristic attractions, hotels or restaurants; whereas for taxi trajectories [14] they can correspond to taxi stops, and common cycling or jogging points for cyclists and runners trajectories [12][3].

4.3. Results related to users similarity

In section 3.5 was described an approach to calculate similarity between users by the number of times each pair of individual have visited the same region. Figure 13 shows the result table

displayed by our visualization tool with the most similar users on the top. Additionally, the tool allows filtering the result by users, and sorting them by their similarity.

User A	User B	Similarity	▲ %
INDV 0	INDV 3	36/50	72.00 %
INDV 3	INDV 4	35/57	61.40 %
INDV 0	INDV 4	28/52	53.85 %
INDV 1	INDV 5	4/21	19.05 %
INDV 1	INDV 4	9/48	18.75 %
INDV 5	INDV 8	3/16	18.75 %
INDV 0	INDV 1	8/43	18.60 %
INDV 8	INDV 9	2/11	18.18 %
INDV 1	INDV 12	3/17	17.65 %
INDV 4	INDV 5	8/46	17.39 %
INDV 1	INDV 3	9/54	16.67 %
INDV 8	INDV 12	2/12	16.67 %
INDV 1	INDV 8	3/19	15.79 %
INDV 0	INDV 5	6/42	14.29 %
INDV 5	INDV 9	2/14	14.29 %

1-15 of 45

Figure 13. Users' similarity table.

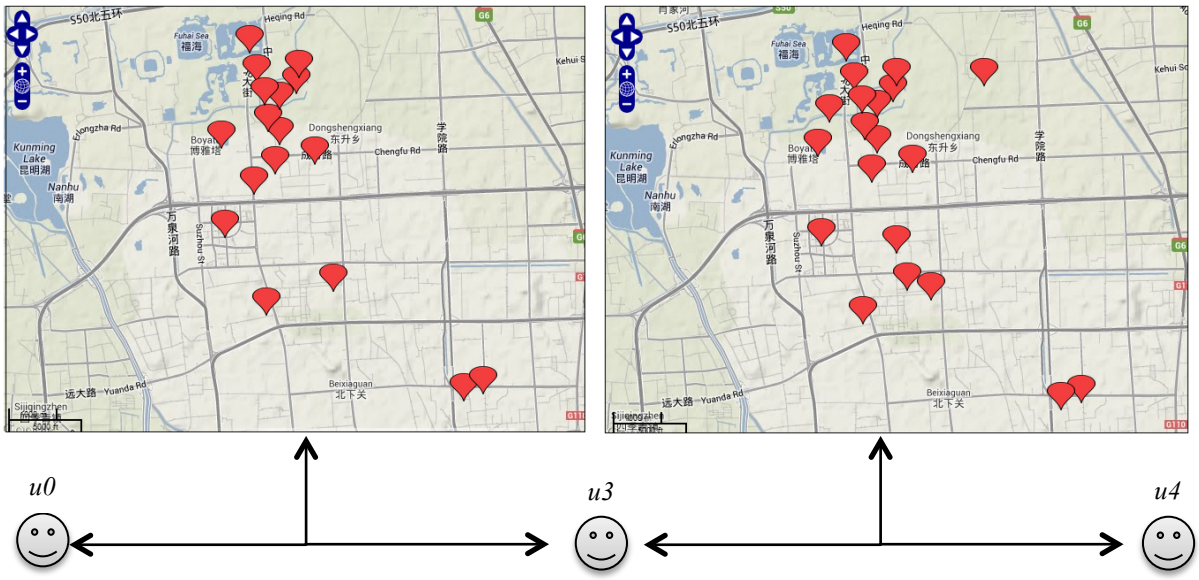


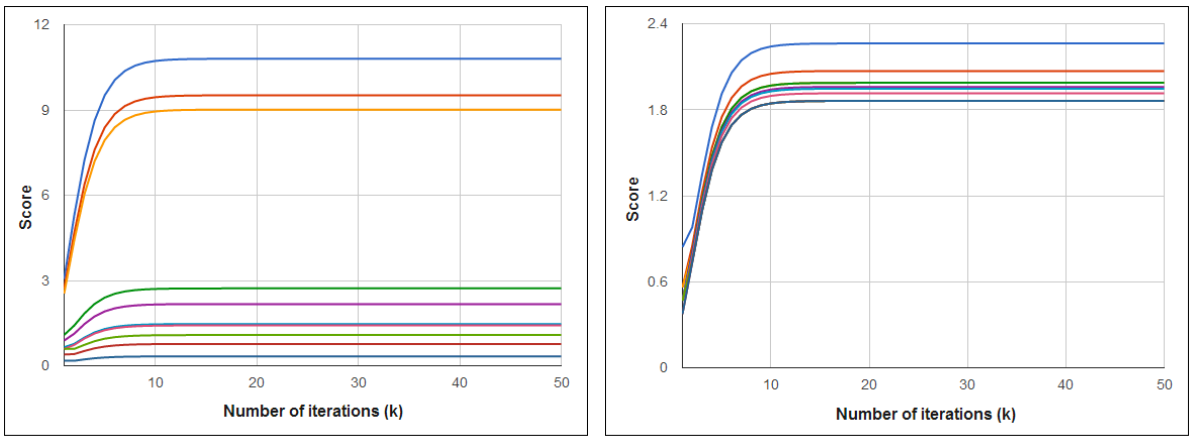
Figure 14. Locations visited by pairs of individuals.

The column *Similarity* contains the similarity product *Sim(u1, u2)* as described in section 3.5. Figure 14 illustrates two maps with some of the locations visited by the two most similar pair of individuals (i.e. *u0* and *u3*, *u3* and *u4*). As one may notice, they share many POIs in a same region. Thus, those individuals with the highest similarity are more likely to know each other and have similar location preferences. For the GeoLife dataset, these individuals can be for instance university colleagues. Therefore, this approach can be useful for friends' suggestion in social networks. Facebook, for instance, suggest friend based on the number of friends two users have in common. Similarly, this approach can be applied in the same manner in a social network based on geographic location.

Furthermore, this strategy can be applied for different GPS logs or other source of geographic information from users, with the purpose of finding individuals' similarity based on their geographic location record, stay points or points of interest.

4.4. Results related to users and locations ranking

In this section will be discussed the results for the HITS-based model presented in section 3.6. Figure 15 illustrates the final scores for the top 10 authorities and hubs after running Algorithm 3 for users and POIs for 50 different values of k (number of repeated iterations of the update steps). One may notice a convergence of both scores for $k > 10$ iterations, for this reason in this experiment the value for $k = 10$ has been adopted.



a) Hubs score convergence b) Authorities score convergence

Figure 15. HITS score convergence

The final score tables for the top 10 hubs and authorities, displayed by the visualization tool, are shown in Figure 16.

Rank	User Id	Score
1	INDV 3	10.79713
2	INDV 4	09.51151
3	INDV 0	09.00745
4	INDV 1	02.71977
5	INDV 5	02.16906
6	INDV 8	01.46620
7	INDV 12	01.41876
8	INDV 2	01.07923
9	INDV 9	00.76561
10	INDV 6	00.32940

1-10 of 10

a) Top 10 Hubs

Rank	Point Id	Score
1	POI 1999767	02.26111
2	POI 1999780	02.06883
3	POI 1999764	01.98643
4	POI 1999791	01.98643
5	POI 1999796	01.95810
6	POI 1999803	01.94561
7	POI 1999797	01.91363
8	POI 1999766	01.86047
9	POI 1999772	01.86047
10	POI 1999789	01.86047

1-10 of 10

b) Top 10 Authorities

Figure 16. Hubs and Authorities

Ranking is useful for understanding both users and locations. Users with the highest ranking are more likely to know a certain region better, while the highest ranked authorities are more likely to be the main points of interest. This is a good approach for location suggestion and detection of experience users. For the GeoLife case, the hubs are those individuals in the database more likely to know the university campus or the city of Beijing better; authorities in turn can be locations such as the main university buildings or leisure locations.

5. Conclusion

Geographic data mining has become an attractive field of study due to the increasing number of access to electronic devices, which can record user-generated GPS trajectories. This paper has shown how raw GPS data can be managed by means of data mining techniques to extract useful and interesting features from trajectories and users. Such features include Points of Interest, which are regions that one individual, or many, have spent a considerable time on its surroundings. This work has described the applicability of POIs to understand both users and locations, and as a consequence support users to make decision.

Furthermore, this paper presented how to calculate individuals' similarity based on their location history, which is useful to understand individuals' interaction in a certain region. Later, a HITS-based approach to rank both users and locations was conducted to infer experienced users and main locations of interest in a region. Besides, a simple Web tool has

been developed in this project to support the visualization of the results, allowing a better understanding of the achievements.

The results presented here are effective to understand a society based on its individuals geographic movement over the time, and help individuals to make better decisions. Future works include the improvement of the data cleaning method using machine learning and statistical techniques; calculate user's similarity based on the time in addition to the location; apply this approach for different datasets and extend our approach to extract paths of interest and common routes among locations, to support individuals when moving to an unfamiliar location.

As a personal reflection of this project, it was quite important to be able to learn and present how geographic data can be managed and how some data mining techniques can be applied in this specific field to retrieve interesting features. As a personal evaluation of the achievements, they clearly show the potential of geographic data mining for understanding individuals and locations, and how these techniques can be applied to understand the society.

Acknowledgments

The Brazilian National Council for Technological and Scientific Development (CNPq) for supporting my scholarship in Australia. The Australian National University and my supervisor Dr. Lexing Xie for giving me the opportunity to study and work in this project and improve my background in researching and geographic data mining.

References

[1] Choudhury, M. D.; Feldman, M.; Amer-Yahia, S.; Golbandi, N.; Lempel, R. & Yu, C. (2010), Automatic construction of travel itineraries using social breadcrumbs., *in Mark H. Chignell & Elaine Toms, ed., 'HT' , ACM*, , pp. 35-44 .

[2] Ester, M., Kriegel, H.P., Sander, J., Xu, X. (1996-). A density-based algorithm for discovering clusters in large spatial databases with noise. *In Evangelos Simoudis, Jiawei Han, Usama M. Fayyad. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (KDD-96). AAAI Press. pp. 226–231. ISBN 1-57735-004-9.

[3] Garmin connect. (2012). http://connect.garmin.com/. *Garmin*. Last access in 20-10-2013

[4] Gudmundsson, J., Thom, A. & Vahrenhold, J. (2012). Of motifs and goals. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* - SIGSPATIAL '12 (p. 129). New York, New York, USA: ACM Press.

[5] Han, J., Kamber, M., Pei, J. (2012) Data Mining: Concepts and Techniques. Third Edition. *The Morgan_Kaufmann Series in Data Management System Series.* Morgan Kaufmann (2012). 550 páginas

[6] Hao, Q., Cai, R., Wang, X., Zhang, L. (2009). Generating Location Overviews with Images and Tags by Mining User-Generated Travelogues. *In ACM Multimedia (ACM MM),* short paper.

[7] Kleinberg, J. M. (1999). "Authoritative sources in a hyperlinked environment". *Journal of the ACM* **46** (5): 604.

[8] Lu, X., Wang, C., Yang, J., Pang, Y., Zhang, L. (2100). Photo2Trip: Generating Travel Routes from Geo-Tagged Photos for Trip Planning, *ACM Multimedia*, Florence, Italy, October 2010.

[9] Microsoft Corporation. GeoLife - User Guide. Version 1.3 (2012/08/01).

[10] OpenLayers (2013). http://openlayers.org/. *OpenLayers: Free Maps for the Web*. Last accessed in 12-10-2013

[11] Rajaraman A. and Ullman, J. D.. 2011. Mining of Massive Datasets. *Cambridge University Press*, New York, NY, USA.

[12] Strava. (2013). http://www.strava.com/. *Strava community*. Last access in 20-10-2013

[13] von Ahn, L. (12-10-2013). "Hubs and Authorities" (PDF). *15-396: Science of the Web Course Notes*. Carnegie Mellon University. Retrieved 12-10-2013.

[14] Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y. (2010). T-drive: driving directions based on taxi trajectories. *In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10).* ACM, New York, NY, USA, 99-108.

[15] Zheng, Y., Zhang, L., Xie, X., Ma, W. (2009). Mining interesting locations and travel sequences from GPS trajectories. *In Proceedings of International conference on World Wild Web (WWW 2009),* Madrid Spain. ACM Press: 791-800.

[16] Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W. (2008). Understanding Mobility Based on GPS Data. *In Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008),* Seoul, Korea. ACM Press: 312-321.

[17] Zheng, Y., Xie, X., Ma, W. (2010). GeoLife: A Collaborative Social Networking Service among User, location and trajectory. Invited paper, *in IEEE Data Engineering Bulletin.* 33, 2, 2010, pp. 32-40.