École polytechnique de Louvain (EPL)

Detecting user's habits using GPS data

Dissertation presented by
Justin LOROY

for obtaining the Master's degree in
Computer Science

Supervisor
John Lee, Michel Verleysen

Readers
Tom HOSTYN, Joachim GIARD, Quentin CAPPART

Academic year 2015-2016

Table of Content

Acknowledgements

This master thesis has been realised with the precious help of my co-promotor, John Lee, to whom I would like to address a special thanks for the precious expertise. Also it would not have been possible without the informed advices of Joachim Giard. He also organized the meetings at the Sony office to keep an eye on my progression through this work, thanks to him. Finally, I'd like to thanks Tom Hostyn for his supervision and for having proposed the subject.

Sincere thank to Professor Verleysen and Quentin Cappart for the time dedicated to the reading of this master thesis.

Finally, I would like to thank my family and friend for their support.

Abstract

Smartphones, smartwatches are some examples of the new connected society. All these devices, the wearables, are following the user in their everyday's life, collecting data for the companies. Localization data is the main subject of this master thesis. The goal is to find a way to extract relevant informations from it. The way it is achieved thesis is by using an algorithm that will output the most repetitive behaviours, called habits, in the list of all trajectories of a user.

Introduction

Problem Presentation

This subject fits right in the now famous big data domain. When in possession of a large amount of data, being able to transform it into real valuable informations is a top priority for any company. Not only because it is a source of income but , on a larger horizon, it leads to a better understanding of complex parts of life and huge progresses in artificial intelligence. This decade brought a new type of technology everybody can enjoy, called wearables (smartwatches, tablets, phones, ...). These connected devices that are used to provide some help to the human's life are in reality a huge opportunity for collecting data, especially location data. Which is now able to be used in a efficient way to extract relevant information. Actually, multiple facets are involved in this data collection. We can find meaningful locations, detect events, determine transportation mode, categorize the user. All these features can be useful for a certain amount of applications in the near future. In this work, the focus will essentially be set on a specific part of the data : the trajectories.

A trajectory is a set of GPS points that represent an user moving from a point A to a point B. Each point is localized in space and time and they are ordered to create a trajectory.

Starting from the trajectories of a user, the goal will be to find some habits of the users. The definition of a habit is a repetitive action in the behaviour of a user. However, there is an obstacle to this objective, that is the complexity and the diversity of life. Indeed, the daily routine of a person is not as simple as it seems.

The work here will not be to solve a classification problem. There is no sense to try to label each trajectory into a habit. That would imply to understand the reason why somebody moves every time he moves. But life is far more complex than that. The algorithm presented here will find among the data of a user, the routes that are often taken by him/her. This is more of a detection problem that will require analysis of the data.

Sony's Project

This subject has been proposed by Sony Techsoft and is part of a bigger project inside the company, which will be presented briefly here. In March 2016, they introduced the Future Lab[10], a structure to allow innovation at a larger level. Inside this lab are developed a set of wearables which will provide a large amount of location data. The goal here, and my contribution to their project, is to make a first attempt at transforming this data into valuable knowledge.

The leader in this domain is Google. The google services installed on all Android devices generate this data so it can know more about its user. For example, the new *Maps* application can show a list of all the trajectories realized in a day. Also with Google Now, a smartphone can become a real assistant. It is one of the first applications of this new technology, it already suggests to start a navigation to some place at times the user usually go to this place.

Unsurprisingly, there are a lot of other companies that use GPS data they collected to improve their knowledge of their clients. This is part of a race to the future technologies everybody will be using very soon.

Literature - State of the art

This domain is approached by many companies and universities, but it is quite large and often seen differently. They are often project-driven. Indeed, the two main papers that served as basis to my work were sponsored by Microsoft or the Massachusetts Institute of Technology .

Location Clustering

Learn Significant Location From GPS[5]

This is a paper from 2003 cited in the bibliography about Placer++ research paper (see below). The aim of this research was to detect meaningful places in the GPS data collected over several users. Therefore, they recorded all trips the team members were doing and ended up with a list of places and tried to cluster them with a custom radius-based clustering algorithm. The same technique will be used in this work, slightly adapted. The reasons for this are covered later.
Once they have all the places, they use a Markov Chain Model to predict sequences in the visited locations.

Mining Interesting Locations and Travel Sequences from GPS Trajectories [1]

This paper is very influential in this domain. They used a dataset of GPS points, called GeoLife, and built a method to extract the two following pieces of information. First, the stay points, the locations where the user stays between the trajectories, and in a second phase, the travel sequences. What is relevant here is the method used to detect stay points.

The data input is the GPS log, a long file filled with consecutive GPS points. To split this list into trajectories, they focus on two situations.

Gap in the GPS signals

If the time between two points is greater than a certain threshold ΔT, this means the trajectory must be cut between these points. In the following formalism, they define **L** as the log of all GPS points, $\mathbf{p_i}$ as a GPS point and $\mathbf{p_i.T}$ the time of the point p_i ,
Traj = $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow ... \rightarrow p_n$
Where

- $p_1 \in L$
- $p_{i+1}.T > p_i.T$
- $p_{i+1}.T - p_i.T < \Delta T$

Stay point

A stay point is a geographic region where the user logged in GPS point for certain time interval without leaving it. The extraction of such points depends on two parameters, the radius of the zone and the time threshold. Such a zone is characterized by a group of consecutive GPS points $P = \{ p_m, p_{m+1}, ..., p_n\}$
where $\forall\ m < i \leq n$

- $Distance(p_m, p_i) < \Delta Dist$
- the time $|p_n.T - p_m.T| \geq \Delta T$

The stay point is centered on the mean position of every points of P. The first point, p_m, is considered as the arrival point and the end of the trajectory. The last point , p_n, is the start of a new trajectory.

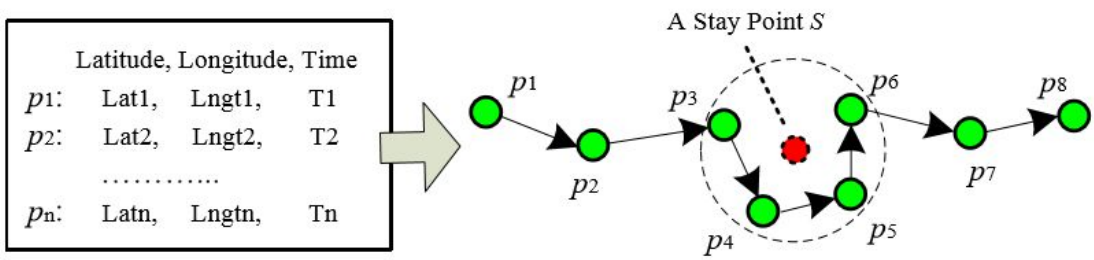


Figure 1 Finding Stay points in GPS log

Habits Labelling

Placer ++ [4]

Placer is a project founded by Microsoft that try to classify places into habits. They have a Machine Learning method to place semantic labels on location. They base their features on innovative concepts. Cross-Labels is the consideration that multiple users can label the same place, but with different purpose. Sequence of labels record what is next place to visit after the current one, and where will the user be 24 hours later. Co-occurrence of labels is the likelihood to see two labels coexist in the same user behaviour. Travel Distance, the one concept that is the most relevant for this work, is the analysis of the average distance the user usually travels to get to a specific place.

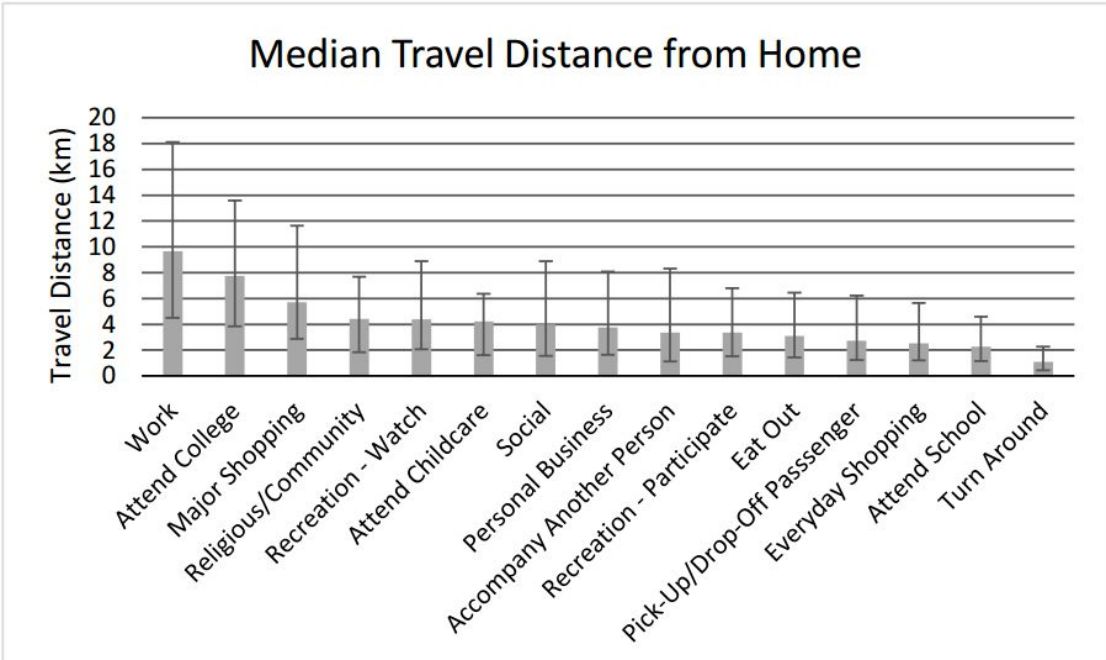


Figure 2 When the starting point is home, this is how far people travel to different types of places. The error bars show the 25th and 75th percentiles.

IDiary[7]

This paper, published in 2015, describes a system to transform GPS data into a searchable database. The goal is to use semantic compression on the trajectories and detect stay points of the user. The results are crossed with an external database to label places and activities. The user can then make queries to retrieve information on where he/she went during the day.

The coreset algorithm they use to compress trajectories has been developed at the MIT. It consists in encoding a large portion of gps points with only a few critical points, (b) on Figure 3, called coreset, that will represent the whole section.

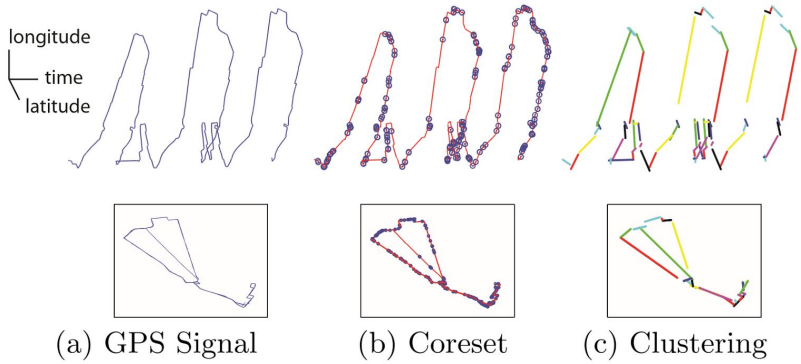


Figure 3 Illustration of the core set algorithm

Data Presentation

Data is at the core of this thesis, especially the points generated by GPS signals. However, there are other sources of data that could bring useful information, like wifi hotspots or NFC chips. These alternative sources are not covered here but are a great idea for future work.

Lynx

Sony Techsoft has developed an app called Lynx. Once installed on a device, it will start a service that collects GPS data every 10 seconds or every 10 meters. Sony has collected data with many members for one year. There is also a website where the user can comment his trajectories. This second phase is useful to create a ground truth and to allow us to compare information we inferred from the data to the reality. We need this to evaluate our precision.

A major concern when it comes to data is the privacy of it. Although every user is totally anonymous, location information are still sensitive and should be maintained behind security walls. For ease in the work and in order to keep the master thesis public, the decision has been made not to use this dataset.

However, the format of this data is relevant to the domain and will be kept so future developers can use the project. Also, this format is needed for Sony's already existing algorithms on the data such as Home and Work detection or Stay detection.

Geolife

The Geolife project conducted by Microsoft aimed at 3 goals :

- Build and share a trajectory life experience
- Find out points of interest
- Make personalized recommendations over friends and locations

This project used a dataset constituted of 1,292,951 kilometers of trajectories collected by 182 users mostly in Beijing over a time window of five years (2007 - 2012). Most of the users are students under 22 years old. The Figure 4 below , taken from the User's Guide of the dataset, plots the distributions of all the GPS points in the city of Beijing. The heatmap shows the density of the points.

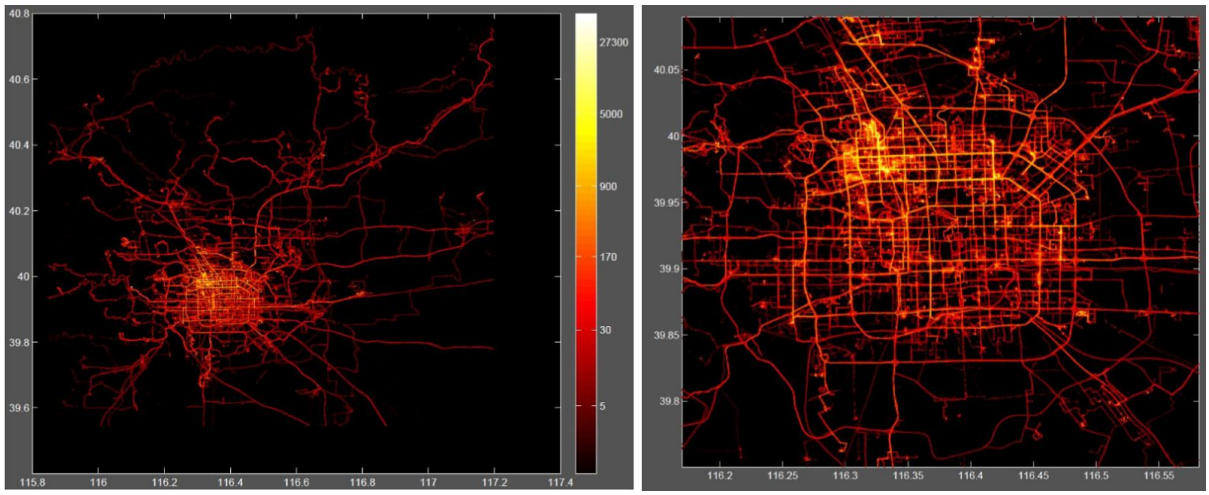


Figure 4 : Heat map of the GPS points in Geolife Dataset

The yellow zone is in fact where the University is situated in the city. It is therefore a strong but realistic assumption to think most of the users were actually member of the university.

This dataset is used rather than the data from the Lynx application, even though they are different in some ways, because it fitted well the context of the thesis and it had less privacy restrictions. Also, the number of users is larger and their trajects recorded during a longer period.

In the Geolife dataset, each user is a directory containing a set of files with .plt extension, that is a list of points representing a trajectory. Every point is represented by a vector of values:

Title	Unit
Latitude	In decimal degrees
Longitude	
[unused]	Set to 0
Altitude	In feet
Date	OLE representation
Date String	yyyy-mm-dd
Time String	hh:mm:ss

Figure 5 Representation of a GPS point in Geolife Dataset

This representation is not very usual, especially for the time representation. The OLE time is the number of days, with decimal, that has passed since the 30th december 1899. Also, the third field is always set to 0 on purpose, without explanation.

Translation into Lynx format

The thesis being realized in cooperation with Sony, the data will be processed by some of their algorithms. In order to stay in the same context, the first operation here will be to translate the Microsoft's database to the same format as the Lynx application. There are some differences that must be filled.

The Lynx dataset consists of one CSV file for each user with all the points registered. Each point is composed of several fields.

Title	Unit
Datetime String	yyyy-mm-dd hh:mm:ss
Datetime String	yyyy-mm-dd hh:mm:ss (the same as above)
Date String	Set to 0
Day of the week	Feet
UTC time	POSIX representation
Offset from greenwich	Seconds
Latitude	Decimal degree
Longitude	Decimal degree
Accuracy	meters

Figure 6 Representation of a GPS point in Lynx Dataset

The major part of the translation is the transition from the time format OLE to POSIX, which is the number of seconds since January the 1st. Some issues are encountered for the offset and accuracy part, which had to be left to 0. Offset is the difference in seconds between the can be computed from the LAT-LONG position, but as the time zone's borders are not easy to find, this part is left for future work. This does not interfere since all the trajectories are located in the same city. Note that the altitude information is lost during the process.

Beside these format details, there are three significative differences with Lynx's database.

Trajectories cut

The data from Microsoft were already processed and cut when the user arrived at his destination. It eased the work a lot, because detecting the places where the user goes was not part of this thesis goals. Sony does already have a stay detection program that would make the transformation of raw data into trajectories, such as it is in the Geolife dataset. Also, this makes the final result dependant of the accuracy of Microsoft's cutting algorithm.

The stay detection problem over this specific dataset has been covered in the paper Mining interesting locations and travel sequences from GPS trajectories[1]. The algorithm will detect a stay point if the user stays in a range of 200 meters during 20 minutes.

No data during stay

As said previously, Lynx uses only one big file with all the points where Microsoft has already cut them into trajectories. But by doing this, we have lost all the points registered when the user does not move. This loss of information is harmfull for detecting the importance of the destination place. For example, there is no way to differentiate a 5 minute stop at a petrol station from a night at home. There is the information of time, but there are sometimes big gaps between the trajectories.

Gaps between trajectories

Something frequently used in the literature is the sequences of trajectories. A user will usually go to the same places in the same order. This supposes to have a complete data of all the trajectories of a person. However, there seems to be gaps in the Geolife Dataset. There can be days, weeks or even months without trajectories and then the user appears somewhere else. This is an indication that the data is incomplete.

Groundtruth

In this database, the users do not have to validate or check their trajectories. So there is no label on them and it is hard to know what the visited places really are. This will later cause some validation problems. Doing it manually is not possible either, as there is no Google Street View data in Beijing or any other precise way to know exactly what are the meanings of some destinations.

When looking at the Geolife data, some problems are encountered in the data itself. Sometimes, there are trajectories starting in the middle of a highway, like if the device recording the position was restarted, or if the user was travelling in a tunnel, or many other causes. I noted also that frequently, a point was misplaced, like if the user teleported himself kilometers away and went back ten seconds later. This noise can occur when working with GPS data and it must be cleaned up. Finally, there is often a lag between the real start and the start of the trajectory, which leaves a massive place detecting problems.

Visualisation

Before getting to the main algorithm, we must have a means of better understanding the data.

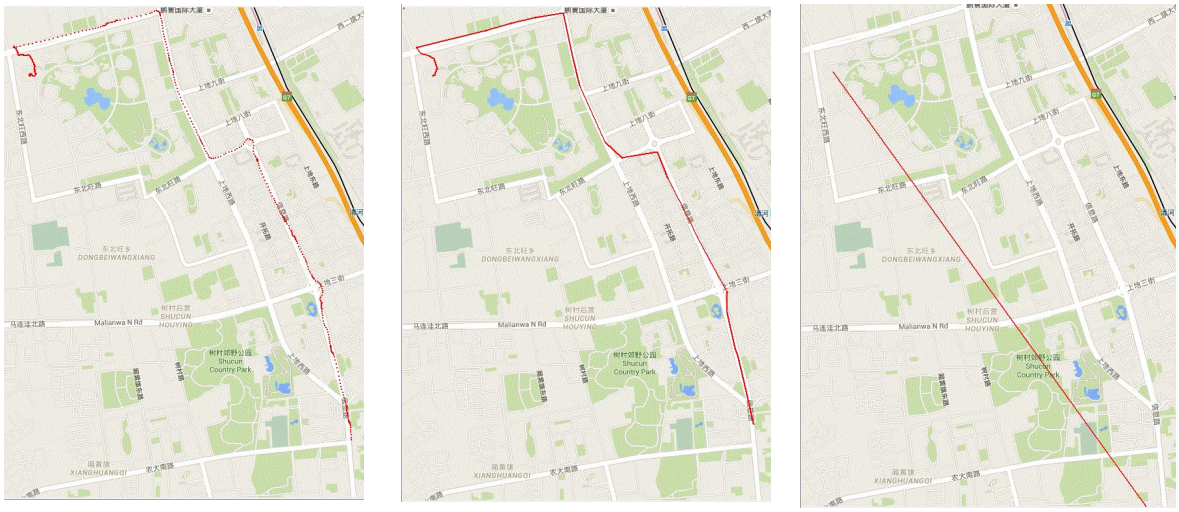
Visualisation Tool

As the human brain can't really process a list of points, It is necessary to have a graphic representation. The first part of the work is to build a program to cope with the visualisation's need. It uses the Zellegraphics Library in Python to draw the trajectories on a map downloaded from Open Street Map or Google Map.

Drawing a trajectory

There are multiple ways to draw the trajectory :

- All the points (it takes some time when drawing a lot of trajectories at once)
- Only a few points and drawing a line between each
- Considering only start and end points



All points | 1 line every 5 points | Line from start to end

Figure 7 Trajectory representations in the visualisation tool

Naturally, the first method is more expensive and can take some time to draw all the trajectories. The last is the fastest but when there are a lot of trajectories, the map quickly becomes unreadable. The balanced solution will often be used to visualize the solutions of the algorithm. However, in this paper, the figures will most of the time be plotted with all the points for clarity purposes.

Drawing multiple trajectories

When looking at all the data from a user, we can draw them on a single map. This is a good visualisation method to determine where the activity is located most of the time. The overlapping trajectories rapidly show frequent used routes. There is also an option to only show start and end points of a trajectory. This will be useful in the future.

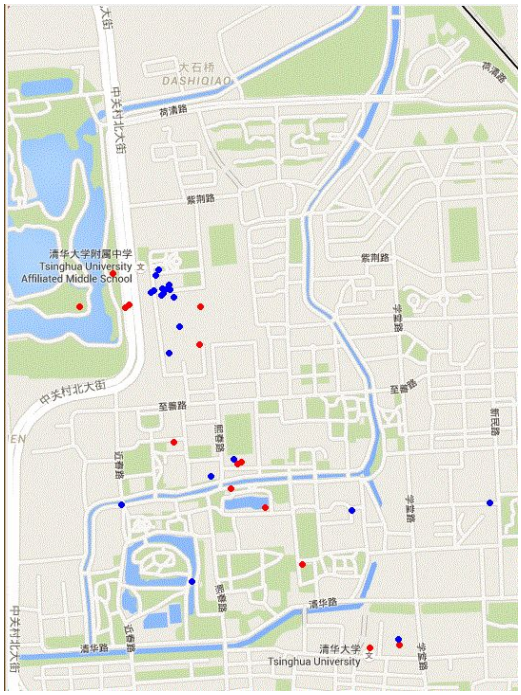


Figure 8 Drawing only **start points** and **end points**

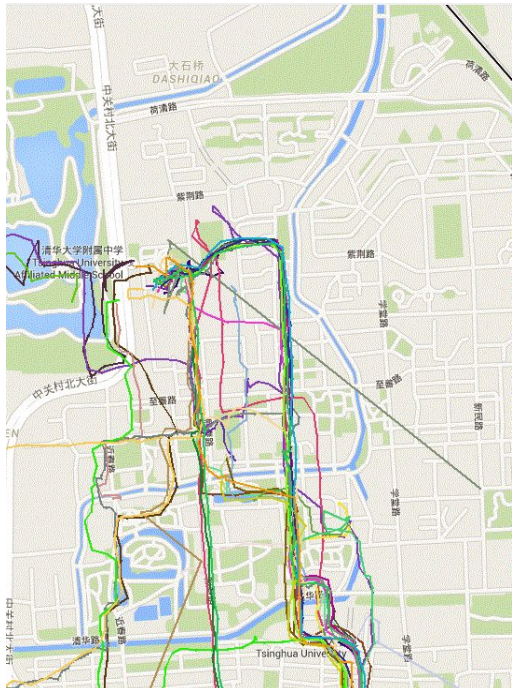


Figure 9 Drawing every trajectory (in random color)

Coordinate Approximation

A recurring problem in geo-localisation domain is the curvature of the Earth. Indeed, when working only with latitude and longitude, the distance represented by this metric can vary regarding where we are situated on earth. A degree of latitude, although the slightly ellipsoid shape of the globe, is always around 111 kilometers. However, the longitude distance goes from 0 to 111.321 kilometers at equator. This can cause trouble to evaluate the speed of a user or to determine the acceptable noise range.

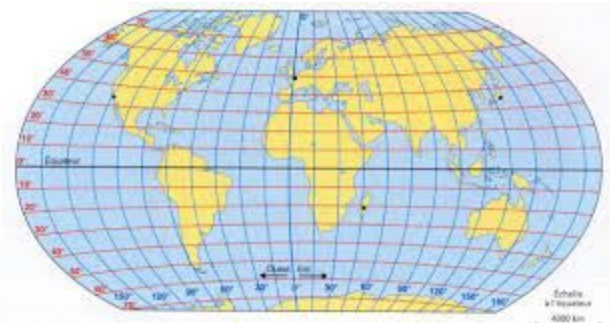


Figure 10 curvature of the earth

Haversine

There is a way to cope with this conversion from coordinate to metric. It requires using the inverse of the Haversine formula. This will compute the distance on the surface of a sphere from the central angle and the radius. However, this formula involves a square root, making it quite expensive to compute between each point.

With ϕ being the latitude and λ the longitude, the distance between two points (ϕ_1, λ_1) and (ϕ_2, λ_2) is :

$$d = 2r\ arcsin\left(\sqrt{sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + cos\left(\varphi_1\right)cos\left(\varphi_2\right)sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

Approximation

In this thesis, all the trajectories are situated inside Beijing. The city covers an area of 0.5 degree in latitude and 0.6 degree in longitude. This affects the longitudinal distance by less than 1%. At the north, one degree is equal to 84.87 kilometers whereas at the south it is equal to 85.69 kilometers. In average, a kilometer is approximated to 0.011726 degree and 1 degree longitude to 85.28 kilometers.

When using this data to compute a distance, we have to multiply any longitude distance by 111035/85280 to have the same metric as in the latitude direction.

Geographic analysis

The example on Figure 11 shows a plot of all the trajectories from a random user, N°42. The map is cropped and zoomed on a zone with some activity.

Here are some numbers describing the user 42 :

- 150 trajectories recorded.
- 239448 GPS points in total.
- 9334 the most points in a trajectory.
- 1596.32 number of points in average
- 4 564 440 meters in total
- 262 648 m the longest trajectory
- 30 429.6 meters in average

He / she is quite an average user.

On the image we clearly see a zone in the North where every trajectory seems to end or start. This place must be significant to the user.

It is easy to note that the road going to the south is frequently used. Some trajectories are more relevant than others. There are many different one.

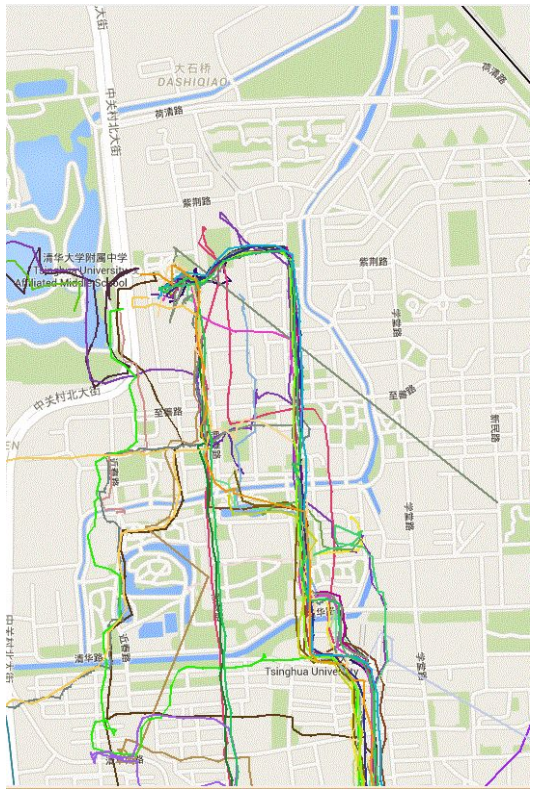


Figure 11 All trajectories from user 42

On the Figure 12, there is a plot of the trajectories from one of the most productive user of the dataset, N°128. From April the 14th 2007 until March the 10th 2011, he has registered 2153 files with 1.208.500 points, that makes 561 points per trajectory on average.

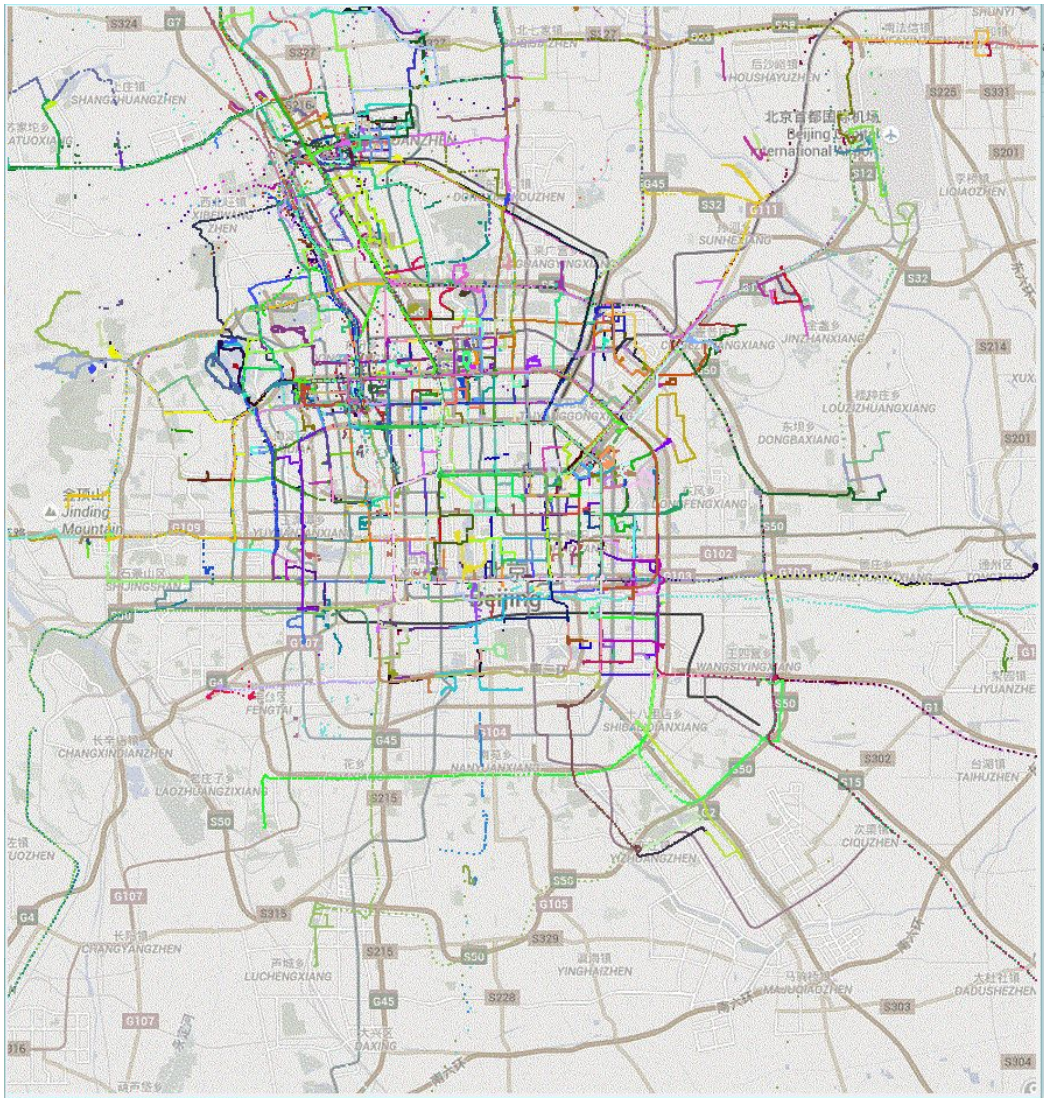


Figure 12 All trajectories from user 168

What this examples shows is surprising. We can infer there can be an enormous diversity in the trajectories of a user. The number of visited places is large, the trajectories go in every direction. Actually, if the map was not shown in background, Beijing would still be recognizable only with the data of this user. Extracting relevant habits in this pool of data is an even greater challenge.

Sony Home-Work detection algorithm

Sony has already developed an algorithm to detect Home and Work places. This information could be crucial in the next steps. To use it, the main problem was the existence of gaps in the data between trajectories. The solution was to fill up the data with points when the user does not move. An easy way is to add a new point on the same place every 10 second if the last destination point is the same as the next start point. After running the algorithm on the data of user N^{r}42, the result was not very accurate.

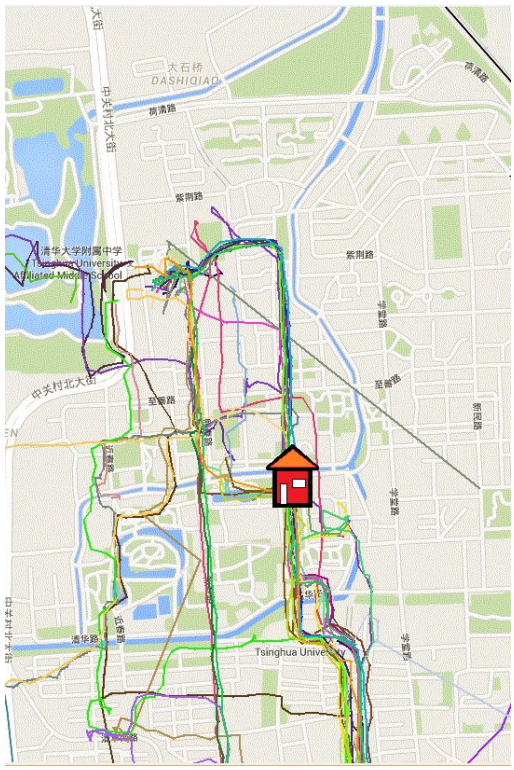


Figure 13 Home detection algorithm

The main cause is that there is some noise in the trajectories, they start later or end earlier, leaving start and end points that does not correspond to a specific place. As shown in the figure 15, that plots all the start positions from the same user, there is a long cluster in the north. This is in fact the road between the Home and the Work of the user. He uses it every day and it should be the most easily recognized habit. However, instead of a clean A to B trajectory, the Geolife dataset is a bit noisy and there are imprecisely cut trajectories that start a bit after the user left or end earlier. The noise is then directed on the trajectory.

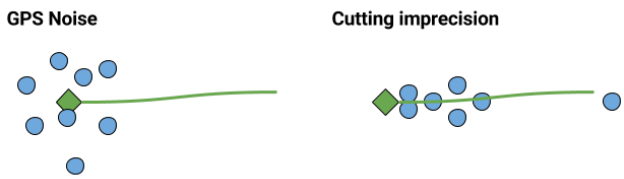


Figure 14 Noise in trajectories. In green is the theoretical start point and trajectory. Blue is the noisy observations.

On the Figure 15, there is a large view of the start points of another user. It shows a strange shape in the red circle. After further investigation, it has been noted that the shape follows exactly the trajectory often used by this user. It can be assumed that it is due to cutting imprecision.

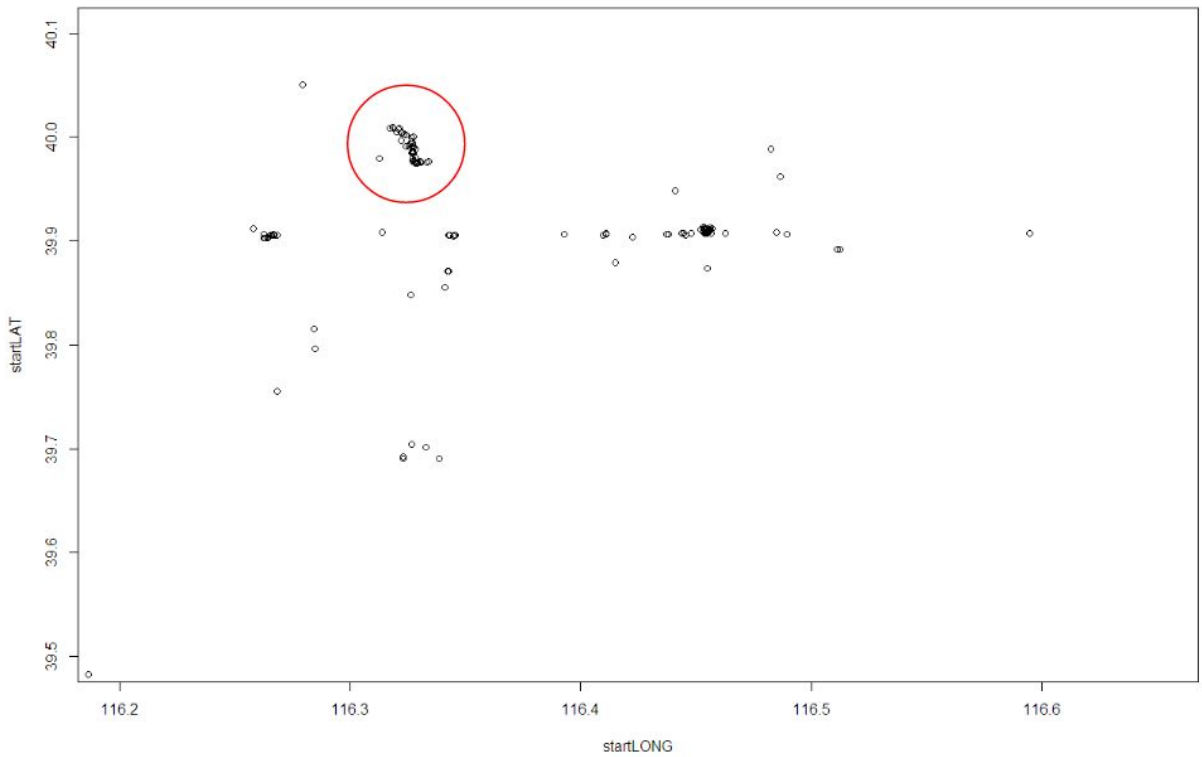


Figure 15 Start positions plot

Data transformation

As seen in the analysis, the data needs a little clean up. Also, working with lists of GPS points is not very effective. Preprocessing the data into trajectory object that would summarize all its properties can improve greatly the performance of the algorithm.

Extracting knowledge about habits from GPS data is done in multiple phases. First the GPS points are transformed into a trajectory object by regrouping in a vector of useful information extracted from the data. Second, a specific algorithm is used to detect relevant places. This will allow to sort the trajectories by Origin-Destination pair. Finally the feature vector is used to regroup the trajectories that look alike. The larger clusters will represent the habits of the user.

Outliers

First, inconsistent data must be deleted. For example, when the user is moving and suddenly he is teleported 1000 meter aside, reaching a speed of 350 m/s, It is safe to assume this is a misplaced GPS point.

Second, some of the trajectories are exceptional and are definitely not part of an habit. The trajectories exceeding 200 kilometers are deleted.

Finally, at a user's level, if there are only a few trajectories, there is probably no good representation of a habit yet. The users with less than 150 trajectories will not be used to test and tune the parameters of the algorithm.

Cutting GPS data into trajectories

The data of the Geolife dataset is already cut between trajectories[1], the method is explained in the introduction chapter.This step is crucial for the quality of the data. An efficient algorithm will be used to detect stay points in the GPS log. This is a challenge that has already been solved by the Sony Techsoft team and is not covered in this thesis.

Feature Engineering

The goal at this step is to build features that are useful and meaningful for the trajectory clustering. The strategy is to conserve a maximum of information from the raw data, while summarizing it in a unique entity. There are various strategies at this point. On one side, the minimum numbers of most useful features are kept. On the other side instead, the accumulation of many features will maybe make the information emerge.

For each file, the algorithm goes through all the GPS points and builds a Trajectory object (Figure 16) that will contain all the features. This vector is then added in a csv file and stored for later use.

Start Point		End Point		Start Time	Duration	Length	Day	Direction
Lat	Long	Lat	Long					

Figure 16 Representation of a Trajectory Object

Aside of this process, the start and end points of each trajectory are gathered to make a list of all places the user went by making an aggregation of all start and end positions. This list is stored in a csv file so it can be reused when needed.

Start and Stop position

The origin and destination is what characterizes the most the trajectory. However, it uses latitude and longitude of both positions to encode this inside the vector meaning it needs 4 dimensions and that is quite expensive. The values are just translated from the first and last points of the file. These features will not be used in the clustering algorithm itself. Instead, it will prepare the clusters by regrouping the trajectories sharing the same origin and destination.

Time, Day of the week

Theses features are intuitively also important to qualify a repetitive behaviour. They are derived from the UTC timestamp in the first point of the trajectory. An important fact to note is that they have to be represented as a circular space. Indeed, 23:00 pm is as far as 1:00 am than 21:00pm.The time is continuous and circular whereas the day of the week is a discrete value, but it has an order. Sunday is before Monday, and Tuesday comes afterward.
The data is fitted on a cycled space using sinus and cosinus to draw a circle following the modulo. On the Figures 17 and 18 we can see two heatmaps over the time in the day (over 24h) and the day in the week (over 7 days) with distances corrected to a circle. On these heat maps, the glow represents the density.

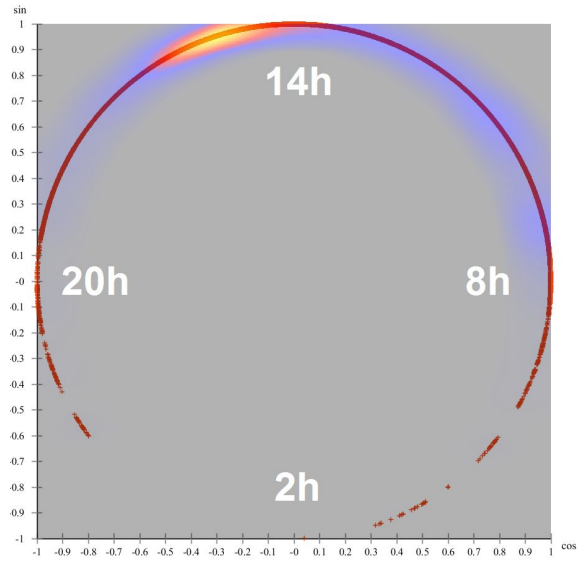


Figure 17 All the points of user Nr42, plotted following the time in the day.

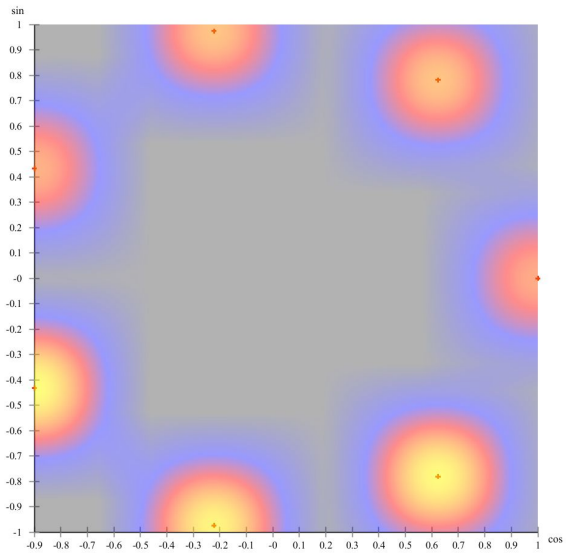


Figure 18 All the trajectories of user Nr128, plotted following the day of the week.

This representation uses two dimensions instead of one. For example, a trajectory made on a Monday 8 pm (12pm UTC) will be represented not by 12.0 (the time) and 0 (the weekday) but by 4 values. But as they are dependent, this is not an issue.

Start Time		Day	
cos	sin	cos	sin
-1.0	0.0	1.0	0.0

Figure 19 Cyclic representation of Start Time and Day of Week

Length and Duration

Classifying trajectories by these variables can lead to unsuspected relation between trajectories. There are different ways to approximate the true length of the trajectory. First, the distances between each point following the trajectory are summed up. However this can lead to a false approximation due to many factors. The noise for example can create a zigzagging travel.

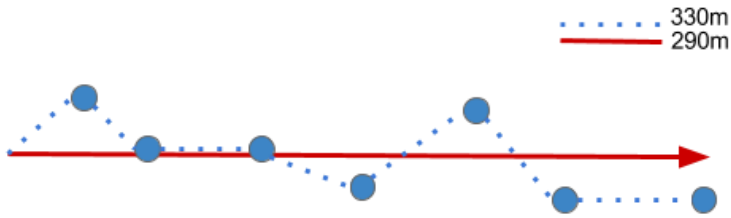


Figure 20 Noise in GPS points

Smoothing the trajectory will give better results in average. It can be achieved by taking only one point every few ones. The more points we omit, the smoother the trajectory will be. But if this number is too large, we can miss corners and end up with an “As the crow flies” distance. Taking one out of three points gave the best results in average.

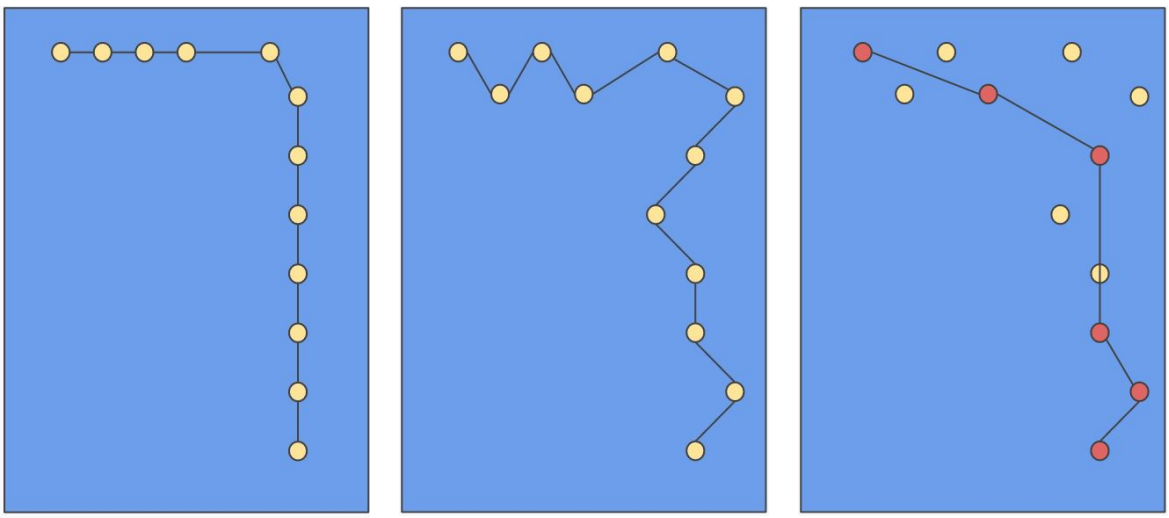


Figure 21 Smoothing the trajectories

Another error compared to reality will come from the approximation on the distance from the coordinates. As said before, 1 degree longitude at the northest point of Beijing is 84.87km whereas 85.69km at South and I approximate it to 85.28km. So an error of 0.5% in average is expected. However, as this error will be the same for every trajectory, it can be ignored.

Actually, length is an important feature to the clustering part. If the start and end points are the same, as in the Figure 22, the trajectory often taken are immediately spotted. Indeed, considering here only the trips between two labels, some are longer and take more time. But we can clearly see a line around 18500 meters that take between 1000 and 3000 seconds.

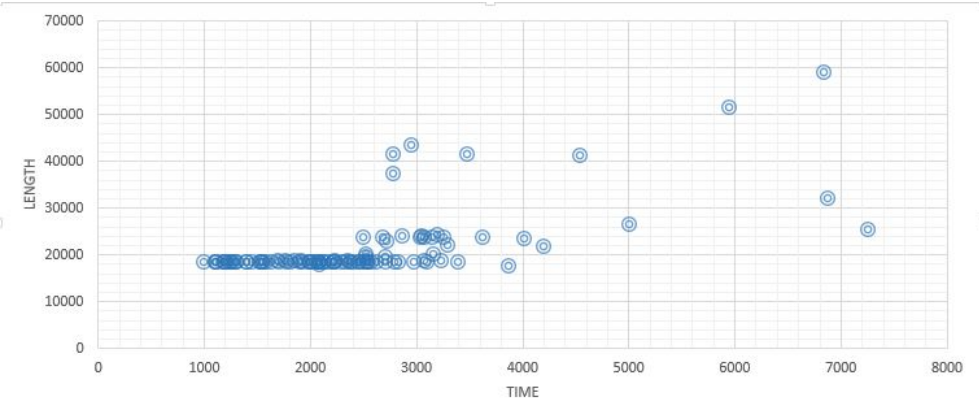


Figure 22 Time and Length of all trips between two labels

Average Direction

From the start and stop position can be derived the average direction of the trajectory. This could help the algorithm to recognize similar trajectories. The angle between the geodesic distance and the latitude line is computed with some simple trigonometry.
To get the angle in radian, one of the formula is

$$arccos(\frac{\Delta Longitude}{Distance})$$

If the $\Delta Latitude$ is negative, meaning the direction is going south, the angle is in fact (2π - angle) because only cosine is considered in the formula.

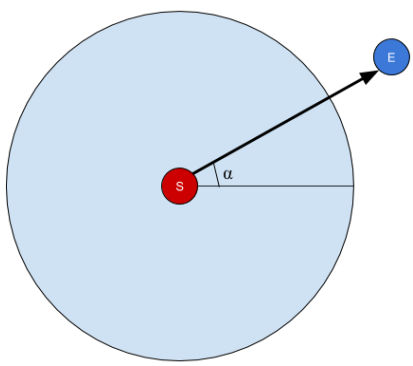


Figure 23 Angle of the direction

Place Labelling

Start and end points

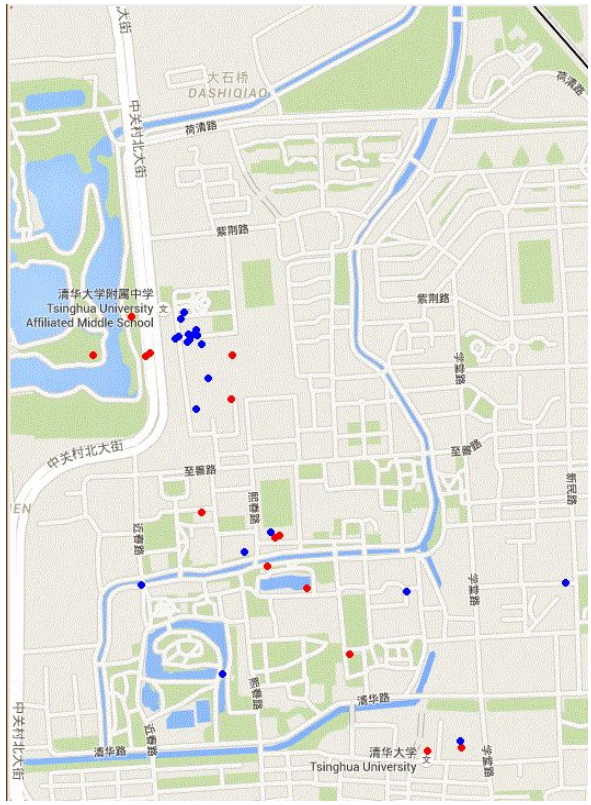
A trajectory is the representation of the user going from a place *A*, the start point, to a place *B*, the end point. These points are detected by the stay point detection algorithm when cutting the stream of GPS positions into trajectories.

Actually, even if sometimes they only represent signal loss at tunnels entries, most of them should logically represent places the user has visited. Thus, in a perfect world without noise, there should be very precise locations. However, as described in the analysis chapter, this is not the case in this world.

On the Figure 24 are represented only the first and last points of every trajectory. Surprisingly, there is a place here where a lot of trips arrive (blue points), but never leave (red points).

For the record, this place is in fact a dormitory in the campus of the Tsinghua University in Beijing. The reason behind this phenomenon is not clear, maybe there is an underground tunnel but it is quite unrealistic. The best guess is that the red points around it are related to this place. But, for a reason, like some noise or approximations in the cutting algorithm, they are misplaced. This again shows how the data can be a bit messy sometimes.

This noise could be reduced by algorithm that will detect cluster of points and regroup them in one average precise location.



●Start points ●End points

Figure 24 start and stop point noise

Detection of important places

As it can be seen on the map, there are places that seems to have more meaning for the user than others because they are visited more often. Knowing the places frequently visited by a user can infer a lot of information about his/her habits.

The idea behind this is intuitive. Considering a habit as a recurrent trajectory between two points, these two points must be visited at least the number of times we observe an occurrence of this habit. The focus should be set then on the most visited places in order to find the most used trajectories.This leads to a pruning of non relevant trajectories that can not be part of a repetitive behaviour because their destination is not often visited.

All the start and end points are merged to build a list of all potential relevant places. Intermediate points are not kept because the user is in movement and this does not help to find important places.

A clustering algorithm, explained in the next section is used to regroup occurrence of points representing the same place.

Then the goal is to identify, among all possible places, the most important ones. Such a meaningful place is called a label. The actual label given to a place is a generic number.

The most dense groups are selected by a threshold of minimum number of observations. This creates a list of labels that represent the most frequently visited places. Labels are drawn with larger orange points.

Start points **End points** **Labels**

Figure 25 Labelling relevant places inside all start and end position

Detection algorithm

They are multiple clustering techniques that can be useful in this situation. The difficulty is to find an algorithm that does not require to know beforehand the number of clusters (like K-means) and that is tolerant with noisy data.
The same problem is encountered in [5], the paper of Ashbrook, D., & Starner, T. (2003), which is explained in the introduction chapter. The decision has been made to use the same custom radius-based clustering technique as they did. However, there is no implementation of this algorithm online. Here are the main steps of it.

Input: **P** a set of points (lat, long), **r** the radius > 0, **t** the minimum size of a cluster and **s**>0 the number of steps in the centering. r, t, s > 0.
Output : **C** a set of clusters

LocationClustering(P,r,t,s):
    C = {∅ }
    While P ≠ ∅ do:
        M = {∅ }
        c = p_i *//chosen randomly* $p_i \in P$
        for 0 to s:
            for $p_x \in P$:
                if distance(c,p_x) < r:
                    M = M ∪ {p_x}
            mean_lat = $\frac{\sum_{p \in M} p.lat}{|M|}$ *//average latitude of the points in M*
            mean_long = $\frac{\sum_{p \in M} p.long}{|M|}$ *//average longitude of the points in M*
            c = new point(mean_lat, mean_long) *//moving the centroid*
            M = {∅ }
        if |M| ≥ t :
            C = C ∪ {c}
        P = P \ M

The Figure 26 is reproduced from the paper where the algorithm comes from. It shows the centering of the centroid in six steps. What they call “location” is called a label in this work, but it is the same concept.

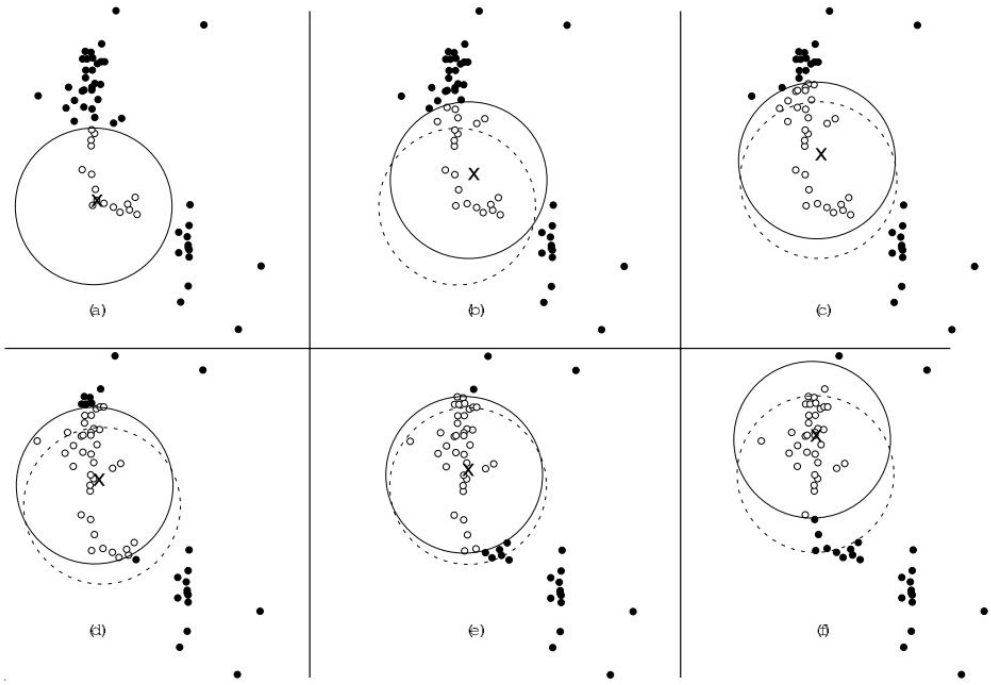


Figure 26 Illustration of radius mean shift clustering algorithm. X is the center of the cluster. White dots are the marked dots. The dotted line is the last location of the cluster border.

This algorithm has been implemented using a Python script. It is capable to find inside the gigantic amount of points small clusters of very dense points. The main parameter is the radius. Even though 20-30 meter seems to be a decent approximation for a radius in a building, the noisy data often need a radius up to 100-150 meter to gather all the start and end points of a specific location. The lack of ground truth makes the validation difficult but after some tests on several users, it seemed to work very well.

This step is decisive for the final results. There can be difficulties when there are big zones larger than the radius with equally distributed points or when the points related to a place have a high-variance noise. This will result into multiple clusters close to each other. This problem can be solved by increasing the radius parameter.

Trajectory clustering

After this step, there will be 3 types of trajectories. First, the ones that goes from a label to a label. That is the group where the focus will be set on. The other two groups, those connecting non labelled places or concerning only one label, are removed from the interesting trajectories.

There are multiple ways to use this information. The solution that has been decided here is to prune all the trajectories that are not connecting two labels. An alternative would have been to eliminate the noise and to center all the trajectories from the cluster to their centroid. This won't help the algorithm but can lead to a destination analysis based on labels for future application. The information about trajectories not often used would not be lost. From this could be derived other type of habits such as "Going back Home, no matter where the user is".

Trajectories sorting

Once the important places have been recognised, the trajectories are filtered. Only the ones going from a label to another one are kept. They are regrouped by the pair of labels they connect. At this step, interesting statistics can be collected like the number of trajectories going from a unlabelled place to another, the ones having one of the end labelled and finally, the trips between labelled places. These numbers give information over the clustering. If the number of label-to-label trajectories is close to zero, it means either there are no habits in the data, or that the clustering it too thin, and the radius should be bigger. After this step, the trajectories are sorted by their labels in a directed way, meaning that the trajectory from A to B is not the same as the trajectory from B to A, even if it is the same route.

Features analysis

Inside every pair of label, there can be multiple routes corresponding to different habits. A second phase of clustering happens here. There are 3 features left that could be used. For demonstration purposes, the different trajectories from label 4 to label 2 from user Nr163 is plotted as example in the Figures 27 - 31.

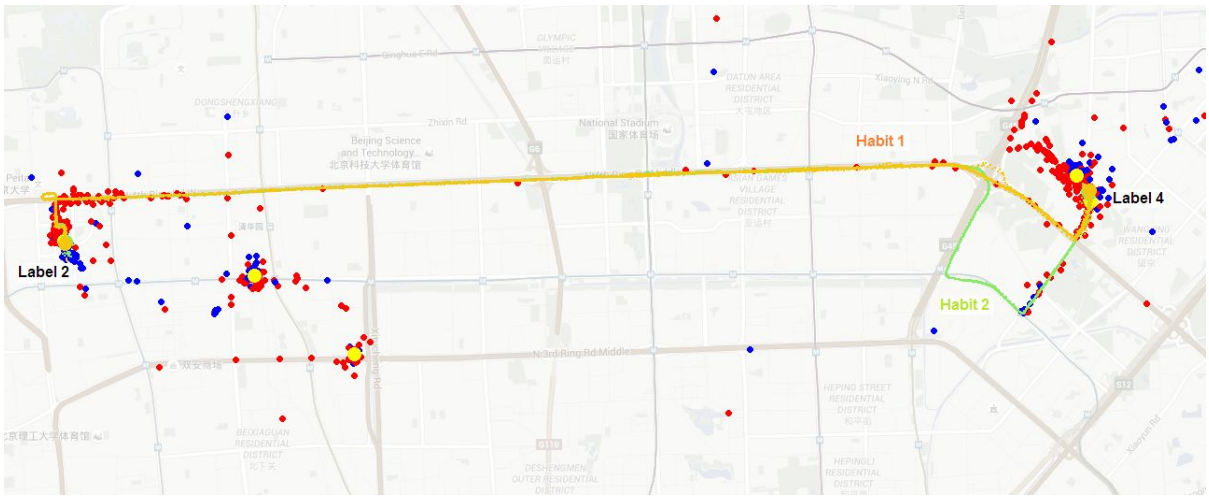


Figure 27 Trajectories between the label 4 and 2 of user 163

Direction

Since all the trajectories go from label A to label B, the average direction is identical. Thus, there is no use for this feature here.

Start time

Intuitively, this indicator would allow to cluster repetitive behaviour that happens at precise time. However, when looking at the data it is quite hard to detect every habit only using this feature.
In the example, one particular time around 6:10 am seems to occurs often.

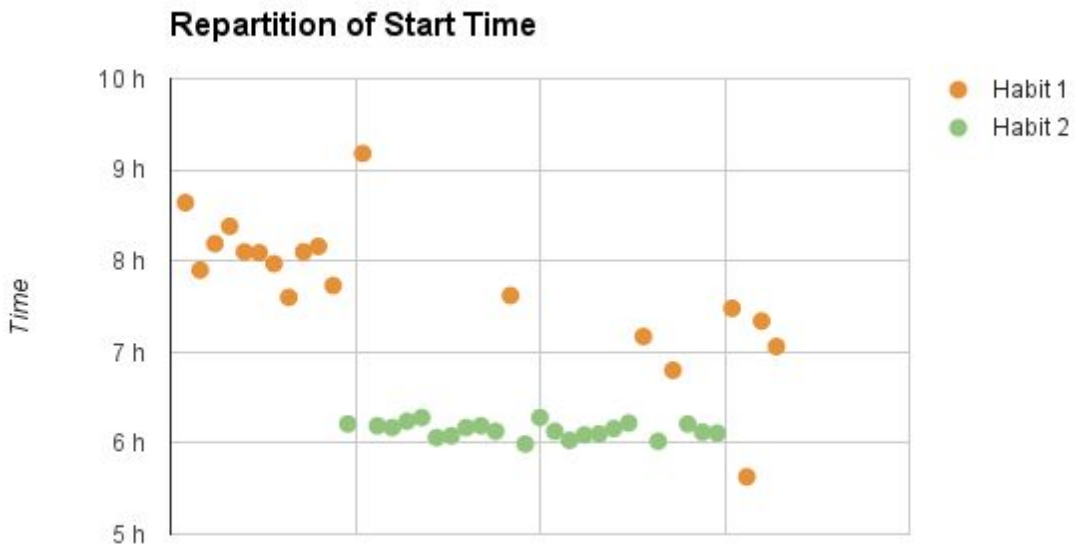


Figure 28 Plot of the distribution of Start Time. Horizontal axis is just chronological order.

Day of the week

This feature does not bring much information on its own, but could be associated with another, like the start time to characterize a repetitive pattern. In the Figure 29, the only interesting cluster to detect is the 4 clusters of Monday to Thursday where there is a lot of leaving around 6:10. The information was already in the start time, but this brings the knowledge that this habit is only present during the week.

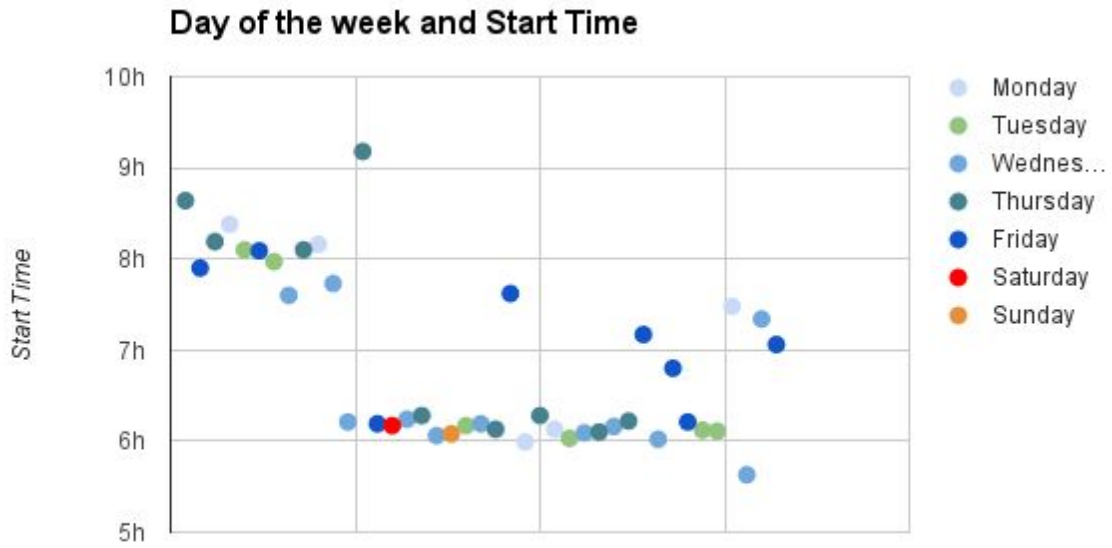


Figure 29 Plot of Start Time in relation with the day of the week. Horizontal axis is just chronological order.

Duration

Actually, time is a very variable value concerning trajectories. As seen on the Figure 30, the duration is very noisy and can't allow a precise detection of cluster.

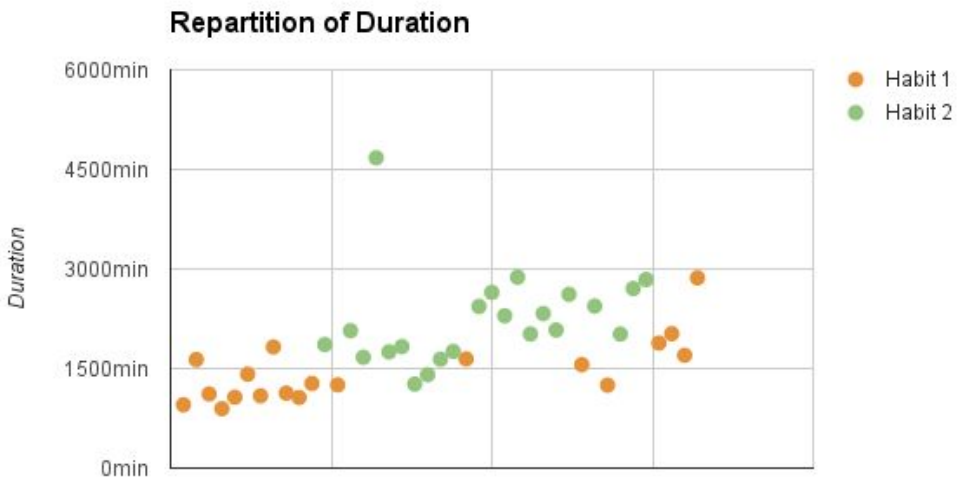


Figure 30 Plot of the distribution of Duration. Horizontal axis is just chronological order.

Length

As seen in the analysis, length depends of the route taken, and does not vary a lot. This allows the algorithm to identify precisely which road has been taken. This leads to a more precise clustering. In the following example we can distinguish two routes, one averaging at 16,8 kilometers and the other around 15 kilometers.

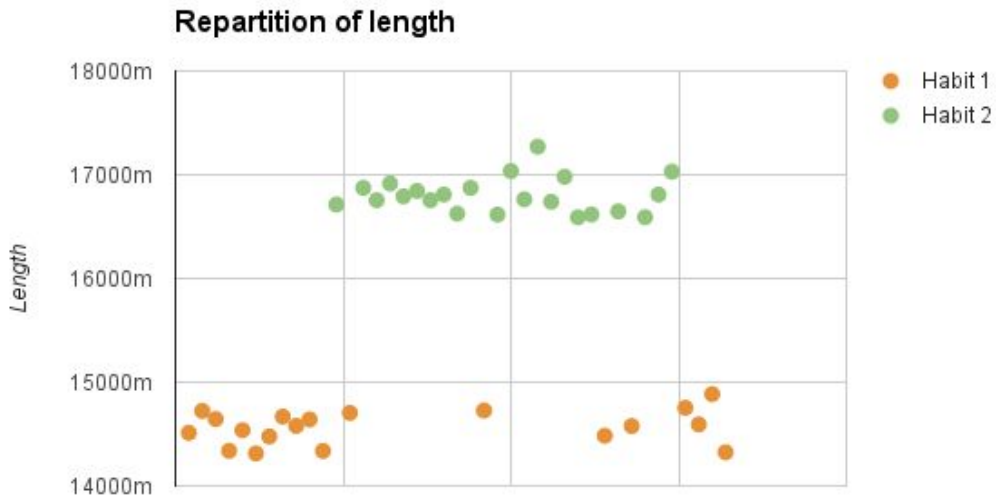


Figure 31 Plot of the distribution of Length. Horizontal axis is just chronological order.

Clustering technique

After the feature analysis, it is explicit that the most easy way to differentiate the habits is the distance. We will have to perform some form of 1D clustering. If the place labelling algorithm is adapted to an interval with a size (=diameter in 2D) of 6% around the centroid, the result obtained is the frequent routes between the labels of this trajectory.

Input: **T** a set of trajectories(length), **i** the interval > 0, **h** the minimum size of a cluster, **s**>0 the number of step in the centering. i, h, s > 0.
Output : **C** a set of clusters

TrajectoryClustering(T,i,h,s):
    C = {∅ }
    While T ≠ ∅ do:
        M = {∅ }
        c = t_i *//chosen randomly* $t_i \in T$
        for 0 to s:
            for $t_x \in$ T :
                if difference(c,t_x) < i :
                    M = M ∪ {t_x}
            c = new point($\frac{\sum_{p \in M} p.length}{|M|}$) *//moving the centroid*
            M = {∅ }
        if |M| > h:
            C = C ∪ {c}
        T = T \ M

6% has been found by looking at the data. The vast majority of trajectories in a habit does not exceed this threshold without using a different route. There is a slight variance in the distance due only to the noise in the trajectory points used to compute the length. In rare cases, the route is different but is coincidently the exact same length. This issue could be avoided in future work by implementing the coreset representation presented in the paper about iDiary[7].

The output consists of groups of trajectories that look alike in terms of start and end point and in length. According to visual assessments, these groups are likely to contains occurrences of exactly the same trajectories. However, it happens some routes only contains only a few occurrences, and some are even unique. The behaviour needs to happen a minimum amount of times to be considered as repetitive. In the algorithm, the threshold is set to 5 trajectories at least.

Minimum habits

At this step, the algorithm has returned a set of habits, consisting of frequently used routes between often visited places. However, the result all depend on how the places were labelled in the first place. In order to evaluate its quality, the following ratio is used.

$$\theta = \frac{Number\ of\ trajectories\ that\ are\ part\ of\ an\ habit}{Number\ of\ trajectories\ connecting\ two\ labels}$$

If θ is low, this means the majority of routes between labels are not used enough to represent an habit. This can be explained by a wrong labelling of the places. If the radius is too small or the points spread too far, there could be multiple label tagging one same place. The trajectories are then shared between the labels and the habit could be missed. These labels need to be merged and we can do it by increasing the radius of the clustering algorithm. The whole process is then repeated.

Once θ goes over a certain threshold, the set of habits is accepted. A good compromise between too much differentiation and over-simplifying need to be found. Indeed, this ratio goes up when the radius of the clustering is increased. This is explained because a larger range means less labels, and thus more trajectories between the labels.

Input: **T** a set of trajectories, **P** a set of places, **k** the threshold for θ. $0 \leq k \leq 1$
Output : **H** a set of habits

ExtractHabit(T,P):
threshold = 0
Trajectories = {∅}
H = {∅}

While $\frac{|Trajectories|}{\sum_{h \in H} |h.members|}$ < k do: // θ check

H = {∅}
threshold = threshold + 25
L = LocationClustering(P, threshold) //extract the labels
Trajectories = t ∈ T such that **t.start ∈ L** and **t.end ∈ L**

for (p_i , p_j) such that p_i , p_j ∈ L: //for every pair of label
I_J_Traj = {t_x} such that t_x.start = p_i and t_x.end = p_j
H = H ∪ TrajectoryClustering(I_J_Traj)

The Figure 32 is a schema summarizing the whole algorithm, starting from the Microsoft Geolife Database until the output of the habits.

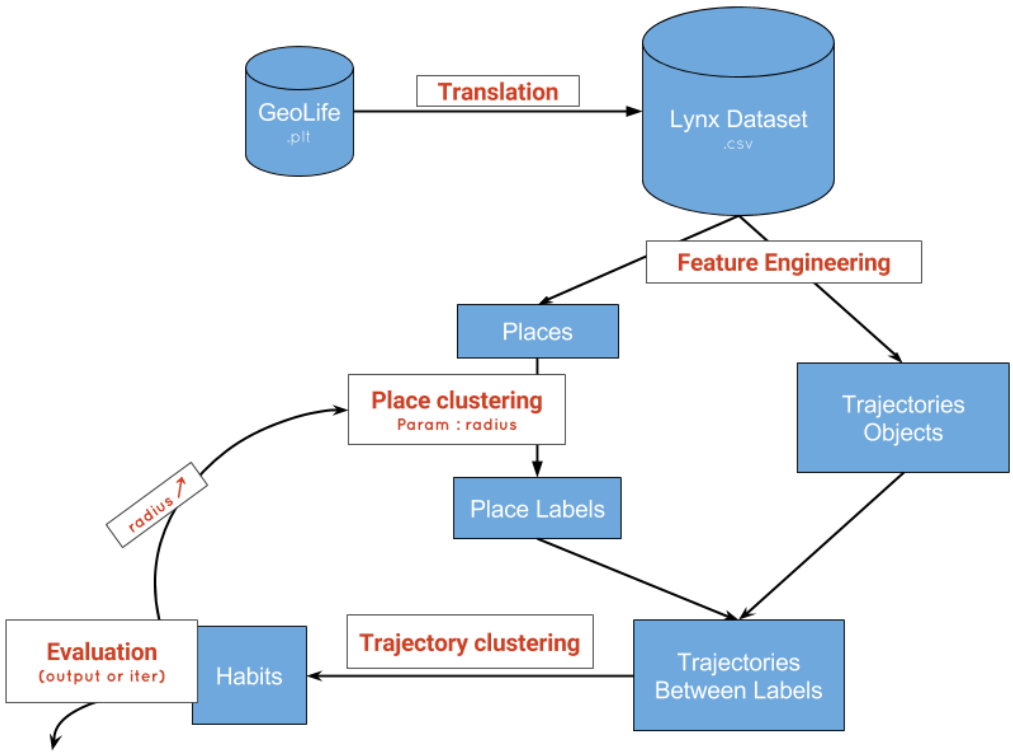


Figure 32 Program Schema

Validation

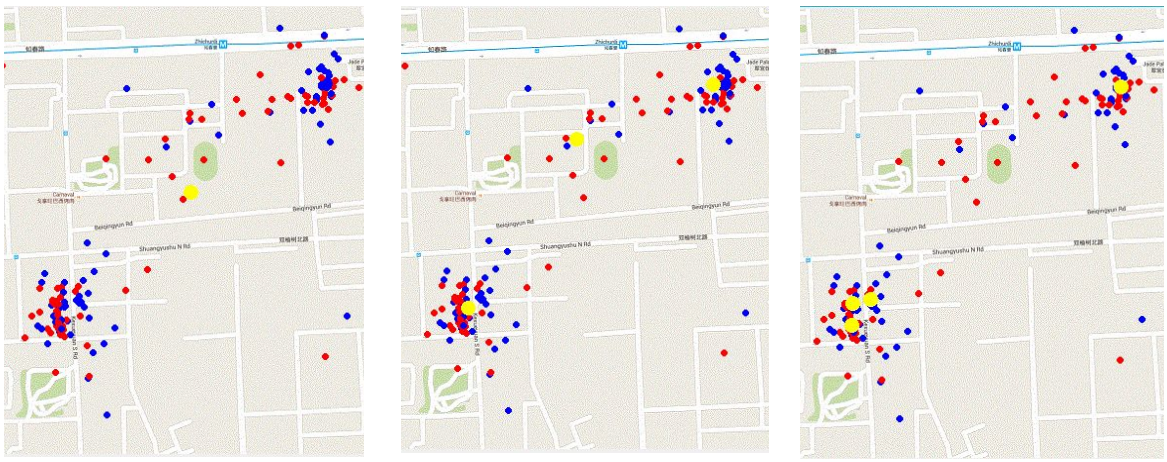
The whole process is made such as it always returns a result, even if it is a very imprecise one such as “Habit from Beijing to Beijing”. But it is hard to evaluate them because there is no ground truth to compare to. In order to cope with this lack of validation system, there are two possibilities. The first is to use a subjective visualisation to estimate the relevance of the result. The second is to insert habits into the data and see if they are detected by the algorithm.

Visual Method

For the first method, the tool made especially for this master thesis is used. There are multiple steps to evaluate in order to verify whether what the algorithm has found is consistent with the reality. The first step to check is the place labelling, then the habits extraction.

Place labelling validation

This is done by plotting all the candidate places, start and end positions of each trajectory, on a map. For a sufficient amount of data, there must be some locations where the density of theses point is higher. Intuitively, these are the relevant places for the user. Following the user, these places can be hard to find sometimes. Because of the lack of data, or the noise that disperses the points all around.



A. Radius 1000m **B.** Radius 350m **C.** Radius 25m

Figure 33 Place labelling with different radiuses.

If the place is missed, like in the Figure 33 A, this means the radius of the clustering algorithm is too large. The points are merged with another location and the center of this cluster is likely to be irrelevant. However, if there are multiple labels for the same place, like in the Figure 33 C, the radius has to be increased. The balance between the two problems will make relevant clustering and should increase the detection of habits.

Habits validation

When the visualisation tool plots the trajectories, a route that is often used will be detected, because of the different colors and the noise making a larger line. Chances are that the habit, if there is one, will pass by this route. However, a large route does not always mean habits, because the user could be using it to go to the majority of its destination.

When drawing the detected habits, the tool draws all trajectories that are part of a certain habit. It is then easy to see if the result goes from a major label to another one.

Shortcomings

There is no objectivity in the visual method, but it can be a quick and easy help when it comes to analyse the data and tune the algorithm. This method is not a good estimator of performance, but ensures consistence.

Concrete testing method

Another way to evaluate the performance of the algorithm without ground-truth is to make one up and test the algorithm on it. This will show how the algorithm reacts to several addition in the data.

Place's Labelling

Once the places have been extracted from the database, it is possible to duplicate or even invent places and add them to the list. This will make some cluster more dense and we can verify then if one of the label is centered over it.

Habits' make up

Invent a habit and adding it to the data is hard to do in a realistic manner. However, duplicating a trajectory enough times can make it enough redundant to be considered as habitual. The test will be to experiment if the algorithm does not miss it.

Creating a new user and making up the whole data is not easy and will not reflect the complexity of life. However, there is a possibility to label by hand the trajectories but this would require to understand the actions of the user. In the end, only a few trajectories are repetitive and eligible for the habit extraction.

Shortcomings

This way of evaluating the algorithm is more objective than the visualisation one. However, it does not ensure the consistency of it and does not reflect the realistic complexity of the data.

Results

The algorithm was run over the data of all 182 users. It founds 254 habits in total, representing less than 4% of the trajectories in average. It is to note that there is a certain number of users that did not record enough trips to be able to deduce some habits.

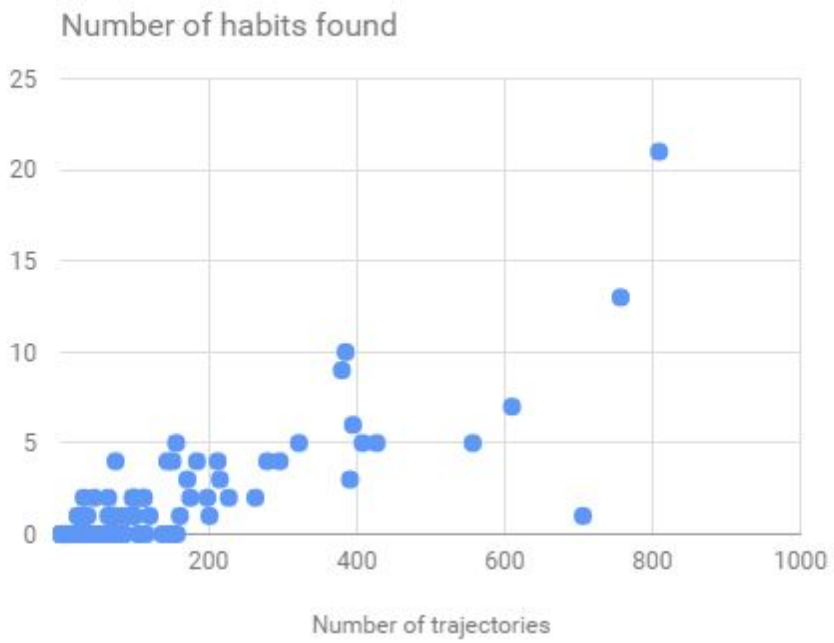


Figure 34 Number of habits found in function of the number of the trajectories in the user's data

Considering only the users with more than 150 trajectories. The algorithm discovered 215 cluster of trajectories. The average number of habits is 7.15, representing on average 10.4% of the data points.

Place labelling

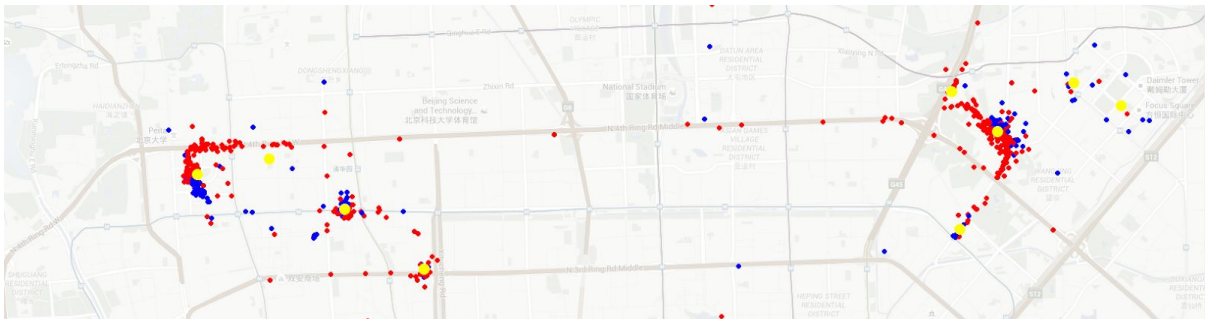


Figure 35 Efficient Labelling example

This is the first and most important step of the algorithm. A good labelling will indicate habits more accurately. It depends mainly on the radius used to check the density. The largest the radius is, the more general the labels will be, meaning a large place will be more easily considered as only one location. However, to stay close to the reality the radius should not be too high. Otherwise, the labelling will be over-simplified and will not be representative of any place. At the end, the only discovery would be that every trajectory goes from Beijing to Beijing.

In the literature, the maximum diameter of a location in this dataset has been set to 200 meters. Here, the habits detection algorithm will try a smaller radius first and then increase it following the final result to improve the quality of the labelling. It will start from 25 meters and go up to 1000 meters with steps of 25.

To validate the method, some of the observation of places are duplicated. For example on the Figure 35 the trajectory ending at the circled point is copied 20 times. The result for the labelling is satisfying, as expected. A label has immediately been detected on this point.

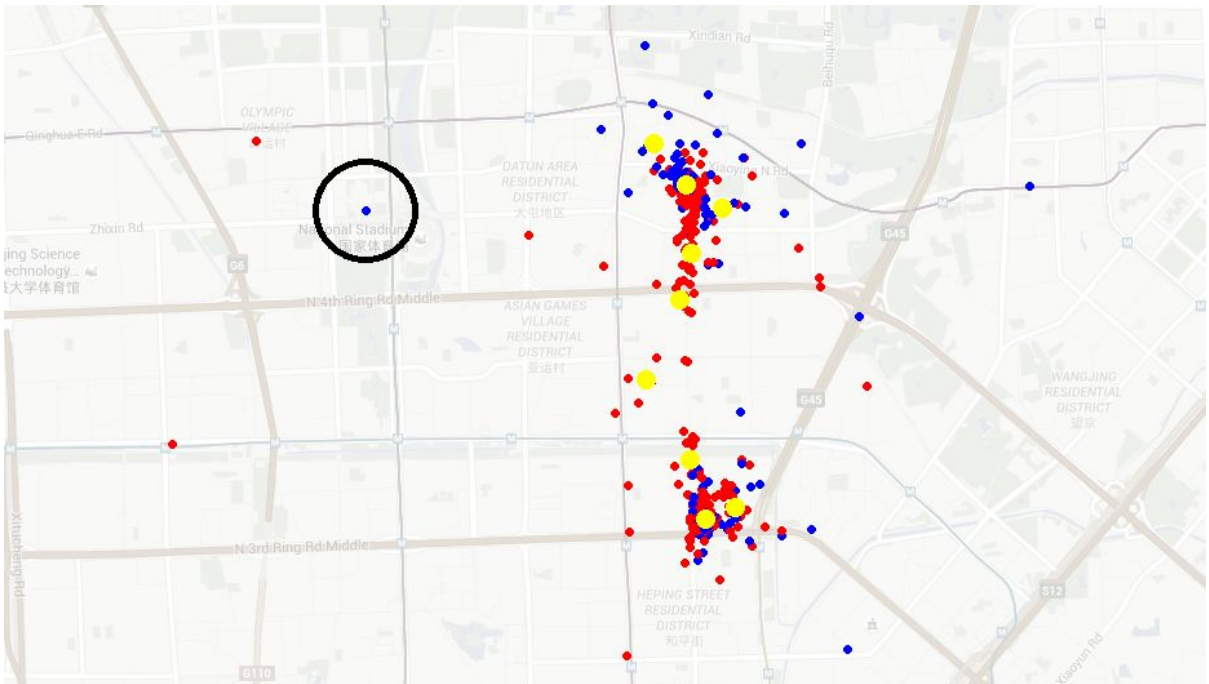


Figure 35 Labels of user N°68. Selection of a place to duplicate.

Habit detection

The output of the program is a list of habits that represents the most recurrent trajectories. When plotted on a map, it is easy to see that the results are coherent and plausible. To realize it, the best way is to look at some examples.

However, to have an objective evaluation,the data injection technique has been tested with habits. Some of the trajectories are boosted in the data to test the program. For example, the same trajectory as Figure 35 is duplicated 20 times. Even though this trajectory seems a bit confuse, an habit has been detected to this place, as it is shown on the Figure 36 A side effect of this data injection is that the θ ratio is also boosted. The place clustering is then satisfying at a smaller radius, making the other label closer to each other.

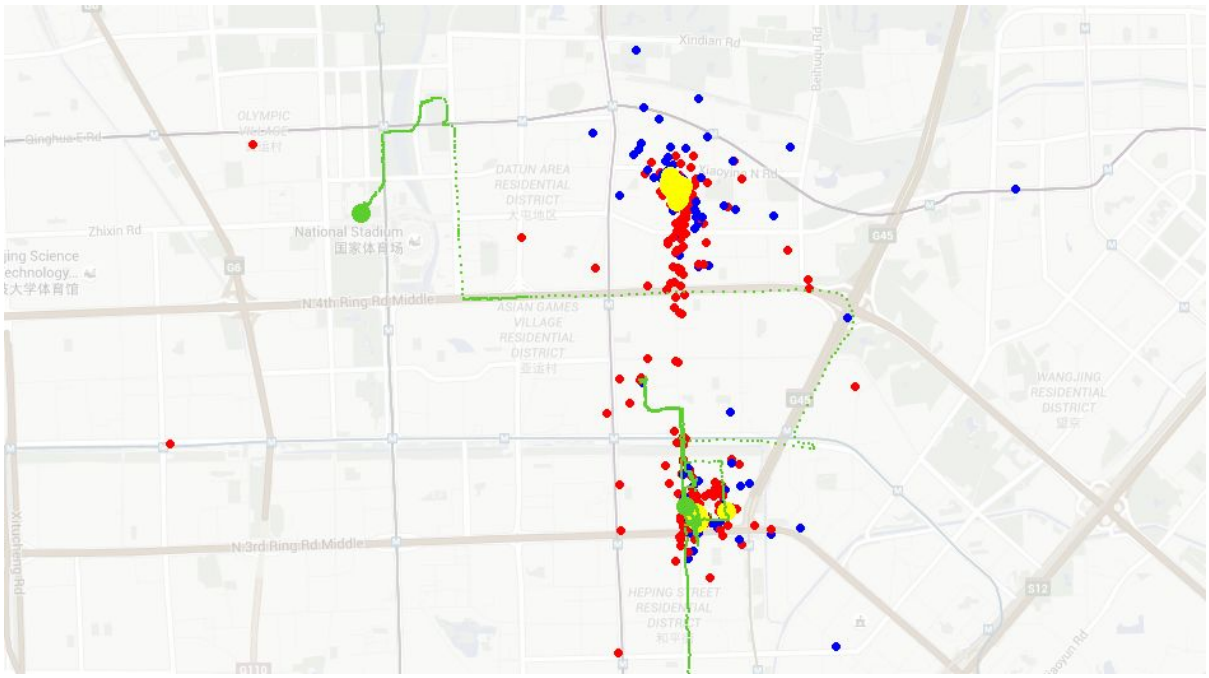


Figure 36 Duplicated trajectory detected as habit.

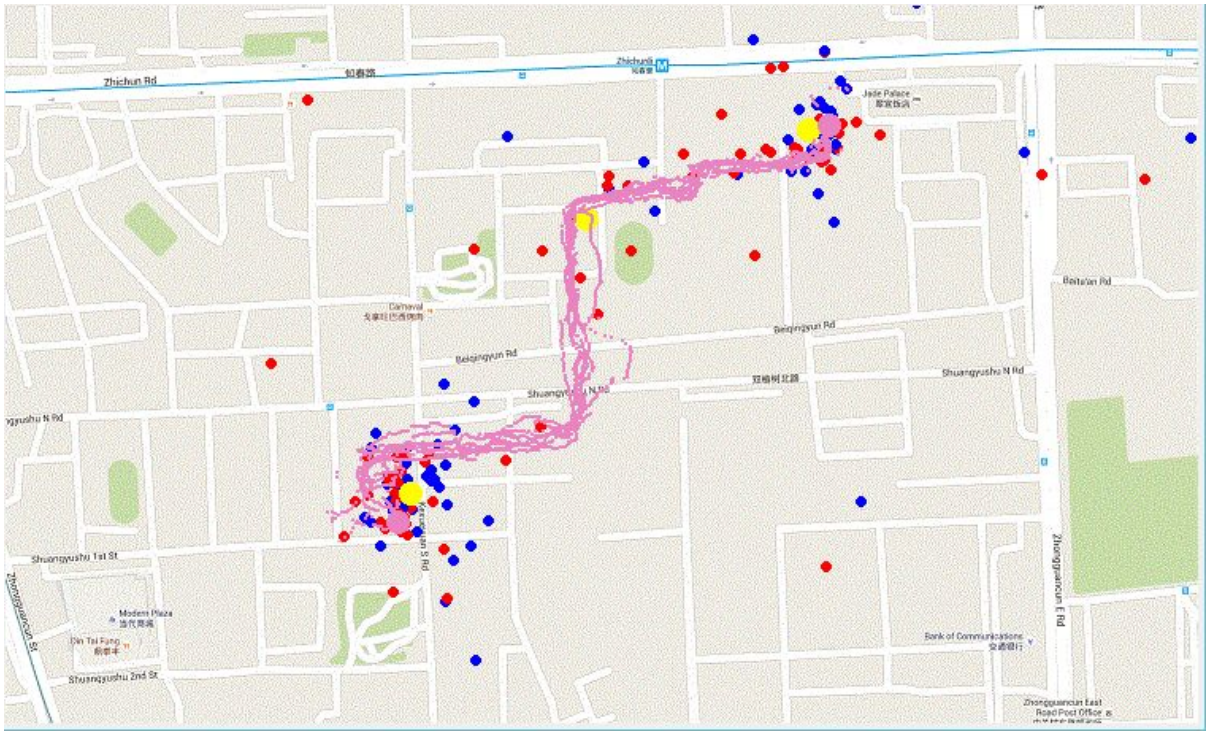
Visual Validation

With visual validation, we can assess that the habits found by the algorithm are often coherent, when fed by enough trajectories. The only way to show this is to look a some examples here below.

Examples of Results

Nr78

This user is located on the campus of the university only. He rarely goes outside this zone, so the points are very close to each other, making the place labelling harder. Although we can clearly see two zones of activity, the place labelling was tempted to gather them in one big cluster. However, the final labelling is fine. The algorithm detected only one habit between the two main clusters.



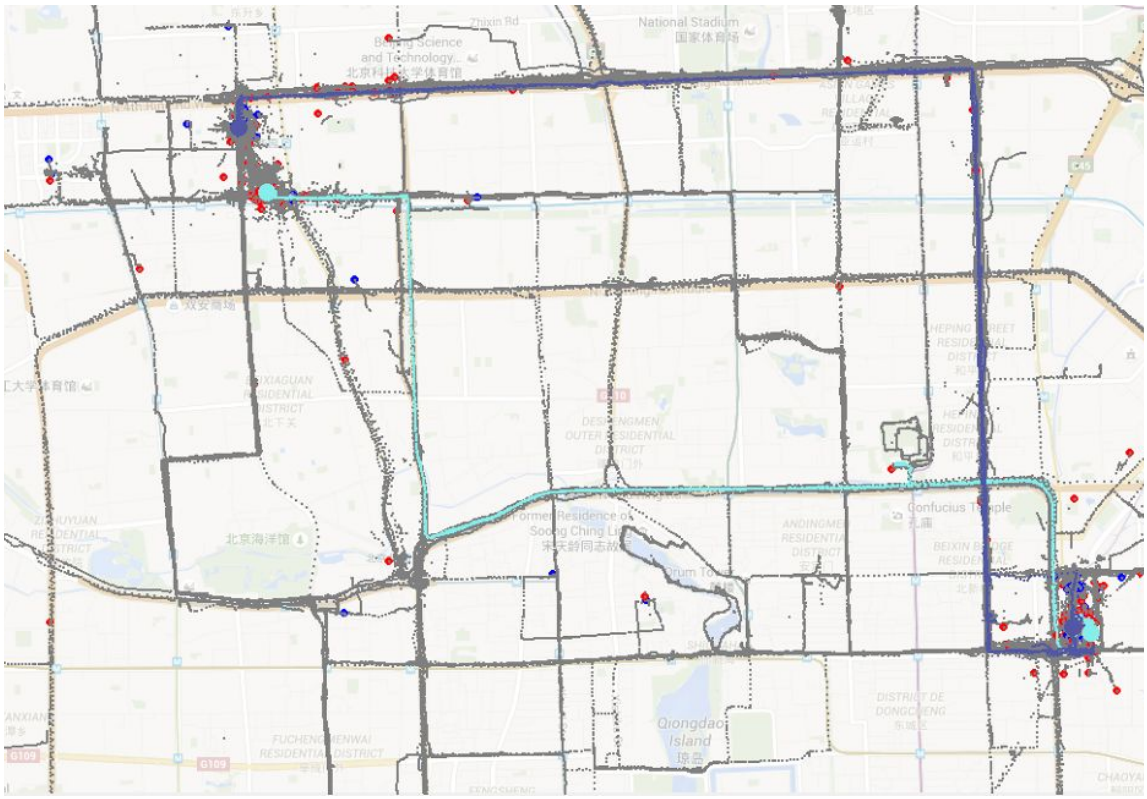
Start/End points of every trajectory. Habit (7 occurences, 1082 m on average). Labels

Figure 37 N°78

Nr84

This user is located in the North of Beijing, There seems to be two main areas where he spends his time and he often goes from one to another. The area are surprisingly large, but we can't know if it is due to the GPS noise or if there is an explanation (like studying in a large university).

On the Figure 38, the two places are represented on the upper-left and bottom-right corners. These places are spread over a certain area, making the labelling inaccurate. Every grey point is part of a trajectory. After running the algorithm, the output was composed of two habits : One from one place to the other, and vice versa.



Start/End points of every trajectory. Habit1 Habit2

Figure 38 Nr84

Nr163

This user has logged in 800 trajectories in total. We can see on his map some clusters of points, the labelling has been made with a 165 meter radius even though the GPS points are often noisy.. The algorithm found 21 habits with 10 occurring more than 9 times and even 4 more than 35. In fact, 23% of his movements are part of a habit.This is a good example of user having a repetitive behaviour.

Figure 39 Nr163

Applications

In the future, knowing this information about a user could create nice applications. We can easily think of a personal assistant that can guess, by crossing the habits with a database of point of interest, a lot of useful information. For example, the opening hours of the stores on the way or the delays on the bus lines following this route.

Destination prediction

This model can be used to predict some of the trajectories the user is going to follow. With a stream of positions, the destination is inferred by checking if the current path is present inside a habit.

Solve Traffic Congestion

In a utopic world, knowing the habits of everybody and be able to predict what are the most probable route everybody will be using could erase the problem of traffic congestion. An artificial intelligence would manage all the cars in a city to suggest a better route to a non-congested alternative.

Future work

Detection of important waypoints

A refinement that could be done to the trajectory object is to find a way to extract a list of important points alongside the trajectory. Theses points could be calculated by the MIT coreset algorithm presented in the introduction. Using these waypoints could allow to merge somes trajectories (like going to a zone).

Alternative sources of localisation

Even if the GPS signals are the best way to localize something geographically, they are not the only one. Even without taking into account other satellite based system like GLONASS, Galileo, COMPASS, there are other ways to know where a device is. For example, location can be inferred from wifi hotspot, IP address (although deprecated), NFC chips installed at home. This additional information could be used in the program to improve accuracy in the place labelling.

Haversine distance

This thesis was centered around Beijing. However, in order to make the program work in the whole world, the computation of the distance between two points only based on their latitude and longitude will need the inverse Haversine formula. The approximation used here is not reliable if we get closer to the poles.

Ethic Question

When talking about learning user's habits using their GPS data, a lot of people can not refrain a little cringe. They all fear that this data could be used against us and they may be right to think so. The Big Brother from George Orwell is not so far away from the Big Data of this world. Scientists know there are questions to be answered about how the data should be used. The technology is evolving at a fast pace and the society is changing too. It is crucial to find a balance between progress and human values. The collection, storage and usage of private information should have legal boundaries. In the paper "*Big Data Ethics*" [8], it is written that there are 4 principles of the good use for big data.

Privacy

Privacy is defined as the right someone has to keep something secret. The information that an individual choose to disclose must be self-managed. However it is not always the case, sometimes information can be inferred without the concerned person's permission. For example, a discount retailer company named Target, in the USA, had an algorithm that detected soon-to-be mother by mining purchases data.The information was then used to send specific advertising to the person. In some cases this leads to unwanted announcement of the good news to some members of the family.

Confidentiality

An information, even if shared with someone or with a company, remains private. The receiver of the information is now in charge of keeping the secret. The fact that this information is collected, even by the government does not make it public.

Transparency

This concept allows the users to know exactly which information about them are possessed by the company and how it will be used. It sets the company and the user on the same level. This is a crucial point to increase the confidence of people in the big institution. This will lead to a better quality of the data and thus to a faster progress.

Identity

If an algorithm can decide what is best for a user, it can cause trouble in the identity of this person. It defines what he/she is even before he/she makes up his/her mind. Not every data inference can be permitted.

I think if these principles are respected, and if the legal structures are created to assure it, the big data society we are constructing is on the right path to progress and technology. The challenge will be not to lose our humanity in this process.

Conclusion

Detection habits in trajectories does not seem at the end like a complex algorithm. However, there were some issues encountered on the way to this solution. The most annoying difficulty was the noise in the data. This make the context very different from a clean, precise dataset. Thanks to the place clustering step, this problem was solved. In second place, the diversity in the trajectories was surprising. In a year of record, some user visited almost the entire map of Beijing. This remark changed the way the challenge was approached. At first the goal was to make a classification algorithm, considering all the trajectories as member of a habit. Then, the decision has been made to build a habit detection algorithm instead. Last but not least, the real issue with this subject was the lack of evaluation method. This caused the work to be subjective, with some arbitrary threshold tuned by hand. This also made it difficult to use machine learning techniques.

The realization of this master thesis is a program that extracts from raw trajectories some meaningful habits. The fact that the results seem plausible is already a success. Furthermore, an in-depth analysis of the obstacles and issues has been conducted. Also, as a bonus, a little visualisation program was developed. It showed its utility during the whole process.

Bibliography

Geolife :

[1] Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma. 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.
[2] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, Wei-Ying Ma. Understanding Mobility Based on GPS Data. In Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008), Seoul, Korea. ACM Press: 312-321.
[3] Yu Zheng, Xing Xie, Wei-Ying Ma, GeoLife: A Collaborative Social Networking Service among User, location and trajectory. Invited paper, in IEEE Data Engineering Bulletin. 33, 2, 2010, pp. 32-40

State of the art

[4] Krumm, J., Rouhana, D., & Chang, M. W. (2015, March). **Placer++**: Semantic place labels beyond the visit. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on* (pp. 11-19). IEEE.
[5] Ashbrook, D., & Starner, T. (2003). Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, *7*(5), 275-286.
[6] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., ... & Theodoridis, Y. (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, *45*(4), 42.
[7]Feldman, D., Sung, C., Sugaya, A., & Rus, D. (2015). idiary: From gps signals to a text-searchable diary. *ACM Transactions on Sensor Networks (TOSN)*,*11*(4), 60.

Visualisation :

Open Street Map : https://www.openstreetmap.org
Google Maps : http://maps.google.com
Zellegraphics : http://mcsp.wartburg.edu/zelle/python/graphics/graphics/graphics.html

Ethics:

[8]Richards, N. M., & King, J. H. (2014). Big data ethics. Wake Forest L. Rev.,49, 393.
[9] Collmann, J., & Matei, S. A. Ethical Reasoning in Big Data.

Others:

[10] http://www.futurelab.sony.net

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve www.uclouvain.be/epl