LIRIS Laboratory

DRIM Group

Internship Report

Transport Mode Detection in the city of Lyon using mobile phone sensors

Jorge Chong

August 17, 2017

1 Introduction

With the massive adoption of mobile phones with a wide variety of sensors (magnetometer, accelerometer, GPS[1], etc), the possibility to capture and analyze mobility data in a systematic way has become an alternative for research. Nevertheless there are some challenges that have to be faced before the adoption of a platform for data collection, like: privacy concerns, user-friendliness for massive adoption, capture parameters adjustment (frequency, what kind of data to collect), power consumption, and finally the deluge of data generated.

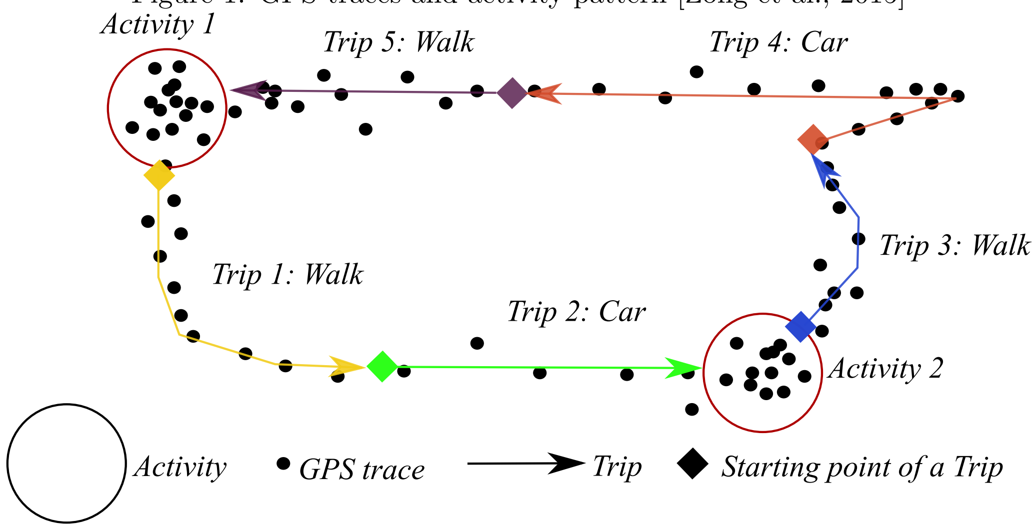
In the context of this project we are interested in extracting the following knowledge from the user mobility data collected:

1. Activities: these are the places where the user carries on activities like: the job place (work), a shopping mall (shopping), a school (carrying children to school), a university (studying), etc.
2. Trips (also called segments in the literature): the movement from one activity point to another using a mode.
3. Mode: this refers to the kinds of transport the user utilized to move from one activity point to another (urban transport, car, walking, etc).

As an example, in Figure 1 we can see a daily pattern of geographical positions (gathered through GPS sensors) and the corresponding trips labeled with their respective transportation modes.

[1] Global Positioning System

Figure 1: GPS traces and activity pattern [Zong et al., 2015]



It is worth noting that many authors refer to activities, trips and traces with different terminology, sometimes using the same word with different meaning. Here we will try to adhere to the one showed in the Figure 1.

2 Context of the Project

The present work is done in the context of the Mobicampus project, funded by IMU. The purpose of Mobicampus is to collect and analyse data about the mobility inside several UDL campuses (Rhone, LyonTech la Doua, Bron, Ecully, Vaulx-en-Velin) with the aim to assists research and decision support activities. The objectives of the Mobicampus project are [IMU, 2016]:

1. Setup a platform for permanent collection of mobility data through web based survey systems.

2. Explore the use of mobile phones as sensors data generation complementary to the survey systems and compare both methods.

3. Knowledge discovery on the data generated about the dynamics of mobility behaviour for future research and decision support.

3 Background

Before delving into the different methods applied in the research literature, it is useful to highlight the characteristics of the sensor data captured and provide some definitions and methods used in processing trajectory data. Mobility research refers to the extraction of high level knowledge from data generated by the activity of moving objects generally acquired through sensors. Researchers would like to understand how, when, where, and why objects move [Renso et al., 2013]. Most of the raw data generated by modern GPS devices is known by the name of trajectories, and refers abstractly to a sequence of locations of a moving object. There are many aspects related to the treatment of trajectories like: collection, processing, trajectories data bases and data mining, techniques to dealing with uncertainty, privacy, and visual analytics. In the following section we will refer to the most important topics relevant to our work.

3.1 Trajectory processing terms definitions

The following definitions come from [Zheng and Zhou, 2011, p. 254]

1. GPS trajectory: a trajectory *Traj* is a sequence of timestamped points, $Traj = p_0, p_1, \ldots p_k$, where $p_i = (x,y,t)$, $(i = 0,1,\ldots,k)$ (x,y) are latitude and longitude, and t is a timestamp. For all i, $p_{i+1}.t > p_i.t$.

2. $Dist(p_i, p_j)$ is the geospatial distance between two points. $Int(p_i, p_j) = abs(p_i.t - p_j.t)$ is the time interval between the two points.

3. Stay Point: s, stands for a geographic region where the user stayed over a certain time interval. The detection depends on a time threshold (τ) and a distance threshold (δ). A stay point s can be considered a virtual location characterized by a sub-trajectory $p_i,\ldots, p_j$ which satisfies the conditions: $\forall k \in [i,j)$, $Dist(p_k, p_{k+1}) < \delta$ *and* $Int(p_i, p_j) > \tau$.

4. Point of interest (POI): It is a stay point with semantic meaning for the user (example: work place, a restaurant, shopping center).

3.2 GPS records

A GPS record refers to a capture of GPS data at some specific moment. Depending on the device, mobile phone GPS sensor chipset and application programming interface, the following are some of the data elements in a GPS record [Gong et al., 2014]:

1. Latitude: Angle which ranges from 0° at the Equator to 90° (North or South) at the poles. It specifies the northsouth position of a point on the Earth's surface [Wikipedia, 2017b].

2. Longitude: Geographic coordinate that specifies the east-west position of a point in Earth. By convention, the Prime Meridian, which passes through the Royal Observatory, Greenwich, England, was allocated the position of zero degrees longitude [Wikipedia, 2017c].

3. Altitude: Height of a point in earth. In GPS devices altitude is measured with reference to a reference ellipsoid (WGS84)[2] that represents the earth's shape [ESRI, 2003].

4. Time: Normally the time of the measurement. Generally it is a Unix timestamp in seconds or miliseconds.

5. Accuracy: an estimation of the accuracy at the location provided.

6. Speed: a calculation of the speed at the moment of the capture.

7. Bearing: the compass direction from the location provided.

8. NSAT: Number of satellites used to calculate the position [Gong et al., 2014].

9. HDOP: Horizontal dilution of precision, measuring how satellites are arranged in the sky at the time of the measurement [Gong et al., 2014].

3.3 Accelerometer data

An accelerometer measures the instantaneous rate of change in velocity [Wikipedia, 2017a], for mobile phone integrated chipsets with accelerometer measures are decomposed in three orthogonal components *(A_x,A_y,A_z)*. The accelerometer sensor data is sampled at frequencies between 30 Hz and 100 Hz [Wang et al., 2010] [Hemminki et al., 2013]. The data generally is:

1. x: Instantaneous acceleration in the x axis

2. y: Instantaneous acceleration in the y axis

3. z: Instantaneous acceleration in the z axis

4. deltaX: Derivative of the acceleration in the x axis

5. deltaY: Derivative of the acceleration in the y axis

6. deltaZ: Derivative of the acceleration in the z axis

3.4 Network data

When a phone call is made or received, we can record the information about the network:

1. Carrier: The network carrier of the device at the moment of the call.

2. Duration: Last call time in seconds.

3. Neighbouring Cells: A List of surrounding mobile cells around the device.

4. Timestamp: The time of the capture.

[2] World Geodesic System 1984

3.5 Wifi data

Because of the proliferation of wifi access points, there is the possibility to obtain wifi network data through a network scan. A network scan is a set of records with at least the following information:

1. BSSID: Basic Service Set Identifier, refers to the MAC address of the access point.
2. RSSI: Received Signal Strength Indicator, it is an indicator of the signal level received.
3. SSID: Service Set Identifier, refers to a unique identifier for a wireless network.
4. Timestamp: the time of the operation.

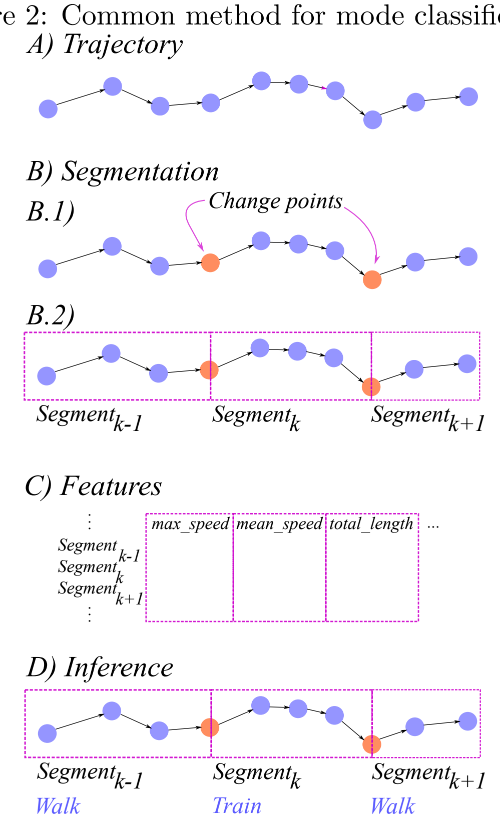
3.6 Data processing

For GPS trajectories, there are many processing steps before attempting to use an inference method. Though not all methods use all the steps, a useful common framework follows the next steps. See Figure 2:

1. Pre-processing: Consists of preparing recorded data, trying to eliminate invalid or useless records or changing the trajectory data to a more precise form:
   - Cleaning based on rules over thresholds (max possible speed in suburban areas, max time of idlying activity, etc).
   - Filtering using different types: mean, median, gaussian, kalman, and particle filters.
2. Segmentation: normally inference methods cannot be applied directly over GPS records, instead given a GPS trajectory we try to divide it into subsets or segments that share the same transportation mode. Referring to Figure 1 a segment whose transportation mode was effectively identified is marked as a Trip and the points that denote the change of transport mode are called Starting point of a Trip, sometimes referred in the literature as "change points" or "mode transfer points". If used, the segmentation process is a critical process because in order to feed the inference stage, features are extracted over the aggregated data in the segment. We couldn't find any clear reference in the literature on the assessment of the sensitivity of segmentation over the inference final results.
3. Inference: the classification is done with feature extraction over a Trip (segment), most commonly calculated on speed, but also with acceleration, heading and accuracy. Different methods used are: thresholds based rules based on domain knowledge, fuzzy logic methods, and machine learning methods. The most common features are max, min, mean, standard deviation, total length and time, percentiles, and other statistical measures.

4. Post-processing: consists of refining the results of the inference phase. Commonly some rules are applied like: any change in transportation mode requires walking before the new transportation mode [Zheng et al., 2008].

Figure 2: Common method for mode classification



For acceleration, because it is a signal sampled a higher rates than GPS locations, the data is normally aggregated over a frame by frame basis using a sliding windows of t seconds, in some cases with overlapping. Figure 3 illustrates the process. Some methods in the literature try to isolate the gravity component of acceleration before aggregating the data [Hemminki et al., 2013] [Mizell, 2003].

The features extracted can be:

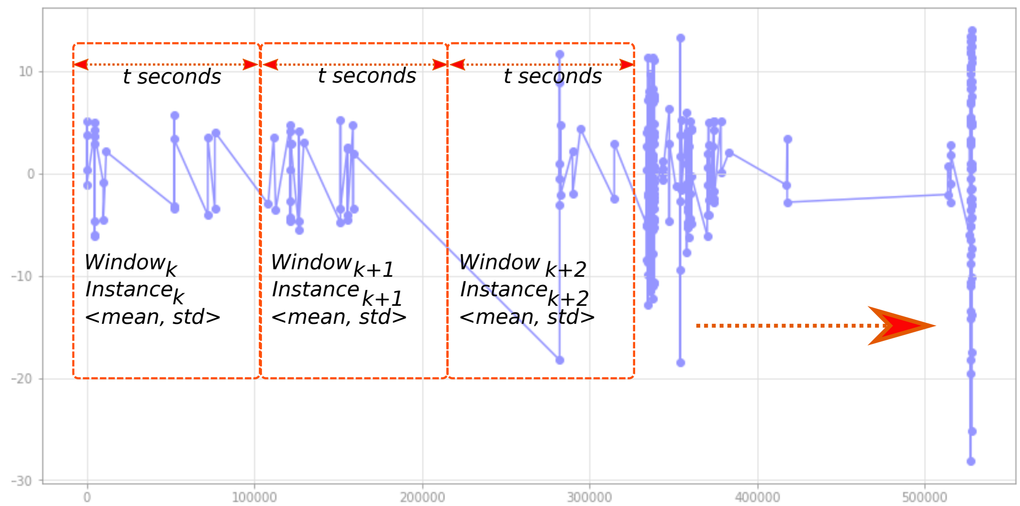
1. Time domain measures, like number of detected peaks over a threshold, zero crossing, autocorrelation.

2. Statistical measures, like mean, standard deviation, Xth percentiles, kurtosis etc.

3. Frequency domain measures, like energy components in specific frequencies using a Discrete Fourier transform (DFT).

The inference is done on a frame by frame basis.

Figure 3: Feature calculations over a sliding window



3.7 State of the art in transportation mode detection

In Table 1 we show the general workflow for each paper analyzed and int Table 2 we see details of the datasets used.

[Zheng et al., 2008] is a GPS based method. It uses GPS loggers and mobile phones to capture GPS data, the rate of capture is 1 capture every second. It uses a segmentation step after some basic pre-processing. The GPS trajectory is subdivided into independent trips if the interval between 2 consecutive points is more than 20 min. The segmentation starts by identifying Non-Walk Segments from Walk Segments in order to find the change point candidates. The algorithm uses parameters like the time between two consecutive GPS records, and the length of candidate segments. The next step is to extract features from each identified segment. The features are: total length, total time, mean velocity over the segment, variance of velocity, top three velocities and top three accelerations, and mean acceleration. Here the authors compare two alternatives: either use an Instance Based classifier like a Decision Tree or use a Conditional Random Field. In the case of an Instance Based classifier the final step is the post-processing in which case the posterior probability for each class is estimated and using a probability matrix of transitions between transport modes, discard implausible transitions like from Bycicle to Bus (This matrix is built from the observed labeled data). In the Conditional Random Field model, there is no need for post-processing, and the features for each segment are fed directly into the classifier.

[Schüssler and Axhausen, 2009] uses GPS data only using GPS loggers. There is no mention on the capture rate. This approach is mostly rules based over speed and acceleration and the use of a fuzzy engine over the speed and acceleration distributions. To do that the workflow follows a pre-processing phase that includes filtering (based on altitude and maximum speeds thresholds between two consecutive GPS records), followed by a smoothing phase using a gaussian filter to remove random errors. The next phase is the segmentation, which starts identifying activities and then further subdividing the segments found into unique mode stages, by identifying change points (the refer to this as Mode Transfer Points). All this is done by following a rules based algorithms. The next step uses a fuzzy engine to infer the transport mode. The fuzzy variables selected are: the median of the speed distribution and the ninety fifth percentiles of the speed and acceleration distributions. A final stage is applied as a post-processing step to assess the reasonability of the final model.

[Bolbol et al., 2012] uses GPS data only with data captured every 60 sec. There is some basic pre-processing step to isolate gaps based on a threshold duration between two consecutive GPS recods. This is a slightly different method. First they consider a "segment" as two consecutive GPS records. The features over each "segment" feed a Support Vector Machine classifier. Second, to have more sample instances they use a sliding window with two thirds overlap over the whole trajectory. The features extracted are: distance, speed, acceleration, difference in heading. A final post-processing step is used to identify the change points.

In [Tsui and Shalaby, 2006] an initial data filtering process is carried out. Then a rules based algorithm for activity and trip identification is applied (based on dwelling time and speeds). After this, there is another stage for identifying the change points (Mode transfer points), again using rules. The mode identification is based on fuzzy logic, using as variables: average speed, 95th percentile of maximum speed, positive median acceleration, data quality. This paper compares two methods: with and without post-processing. The post-processing is done with a GIS (Geographic Information System) and a map matching algorithm to validate the mode identification.

[Dalumpines and Scott, 2017] is also a GPS only inference method, using a specific model of mobile phone and one second capture rate. The pre-processing have a basic filtering phase, and a segmentation phase. Here each segment is an "episode". After identifying the episodes, statistical descriptors are computed over each of them (36 descriptors). The method for mode classification is a Multinomial Logit. The final descriptors used were: median speed, median change in heading, and total duration. There is no post-processing.

[Zong et al., 2015], a GPS only inference method, uses GPS loggers and a capture rate of 5 secs. It follows the common pattern of pre-processing which includes segmentation. The classifier is a support vector machine, and the features extracted for each segment (trip) are: average speed, average acceleration, maximum speed, 75th percentil of acceleration, 75th percentil of speed, travel time, standard deviation of speed, and segment distance. The hyper parameters of the Support Vector Machine classifier are optimized using a genetic algorithm approach.

[Reddy et al., 2010] Uses GPS and accelerometer data. The data is sampled at 1 Hz and 32 Hz respectively, and data aggregated over 1 sec sliding window without overlapping. The GPS data is pre-processed with some basic noise filtering based on accuracy and dilution of precision. For the acceleration, only the magnitude of the instant acceleration vector is taken. The features calculated over the window are: GPS speed, acceleration variance, acceleration DFT (Discrete Fourier Transform) energy coefficients for 1, 2, and 3 Hz separately. For the classifier they tested two methods: a Continuous Hidden Markov Model directly applied over the features, and an instance based classifier followed by a Discrete Hidden Markov Model. This method doesn't do a segmentation step before or after the inference. Instead the output is a stream of classes, one for each of the 1 sec window.

[Wang et al., 2010] Uses accelerometer data. The sampling rate is 35 Hz and data is aggregated over a 8 sec window without overlapping. For the acceleration data processing, first they calculate the magnitud of the acceleration vector, and for the vertical and horizontal axis they estimate the acceleration component after using and algorithm for isolating the gravity component [Mizell, 2003]. The features used are: for the magnitude of the acceleration vector: mean, standard deviation, mean crossing rate, third quartile, sum and standard deviation of frequency components between 0 and 2 Hz, ratio of frequency between 0 and 2 Hz to all frequency components, sum and standard deviation of frequency components between 2 and 4 Hz, ratio of frequency components between 2 and 4 Hz to all frequency components, and spectrum peak position. The same features are calculated for each of the vertical and horizontal components of acceleration. And also the correlation coefficient between these two. In total there are 34 features. The methods applied for learning are: Decision Trees, k Nearest Neighbor and Support Vector Machine.

[Hemminki et al., 2013] Uses accelerometer data only. The sampling rate lies between 60 Hz and 100 Hz. The sliding window is 1.2 sec with one half overlap. This is a hierarchical classification method, which tries to separate first walking from non-walking (kinematic motion classifier). Then, all non-walking identified frames are aggregated into segments and another classification steps are applied. They try to distinguish static from motorized modes (stationary classifier). Once a motorized mean has been detected, the classification of each type of transportation is applied (motorized classifier). Each classifier uses different sets of features (frame, peak, and segment based features), and each of the classifiers uses Adaboost with decision trees with different parameters of T and the size of the trees.

Table 1: Summary of methods used

Paper	Workflow	Modes	Data from
[Zheng et al., 2008] Learning transportation mode from raw gps data for geographic applications on the web	Pre-processing Segmentation Decision Tree Post-processing	Walking Biking Car Bus	GPS
[Schüssler and Axhausen, 2009] Processing GPS raw data without additional information	Pre-processing Trip and Activity detection Mode detection Post Processing	Walking Biking Car Rail Urban Public	GPS
[Bolbol et al., 2012] Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification	Pre-processing SVM over sliding window Post-processing	Walking Biking Bus Car Train Underground	GPS
[Tsui and Shalaby, 2006] Enhanced system for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems	Pre-processing Trip and Activity detection Segmentation Fuzzy inference Post-processing	Bus Car Biking Stationary Walking	GPS and GIS (optional)
[Dalumpines and Scott, 2017] Making mode detection transferrable: extracting activity and travel episodes from gps data using multinomial logit model and python	Pre-processing Segmentation into ”episodes” Multinomial Logit	Stop Walking Car Bus Other	GPS
[Zong et al., 2015] Identifying travel mode with gps data using support vector machines and genetic algorithm	Pre-processing Segmentation SVC Classification	Walking Biking Car Subway Bus Other	GPS
[Reddy et al., 2010] Using mobile phones to determine transportation modes	Pre-processing Feature Extraction over sliding window Decision Tree DHMM	Stop Walking Running Biking Motorized	GPS and Accelerometer
[Wang et al., 2010] Accelerometer based transportation mode recognition on mobile phones	Pre-processing Feature Extraction over sliding window Decision Tree	Bicycle Bus Car Stationary Subway Walking	Accelerometer
[Hemminki et al., 2013] Accelerometer based transportation mode detection on smartphones	Pre-processing Feature Extraction over sliding window Adaboost with trees with depth 2 or 1	Train Bus Stationary Metro Tram Car	Accelerometer

Table 2: Dataset used in research literature

Paper	**Participants**	**Time**	**Place**	**GPS capture rate**	**Labeled data**
[Zheng et al., 2008]	45	6 months	China	1 sec	Yes
[Schüssler and Axhausen, 2009]	4882	6,65 days	Zurick Winterhur Geneva	NA	No
[Bolbol et al., 2012]	81	2 weeks	London	60 sec	Yes
[Tsui and Shalaby, 2006]	9	58 days	Toronto	NA	Yes
[Dalumpines and Scott, 2017]	1967	2 days	Halifax	1 sec	Yes
[Zong et al., 2015]	900	Aprox. 1 month	Beijing	5 sec	Yes
[Reddy et al., 2010]	16	15 min	NA	1 sec	Yes
[Wang et al., 2010]	7	12 hours	NA	Accelerometer based	Yes
[Hemminki et al., 2013]	16	150 hours	Helsinki	Accelerometer based	Yes

Table 3: Acceleration based features

Paper	Features	Capture
[Reddy et al., 2010]	GPS speed Accelerometer variance Accelerometer DFT (3 Hz) Accelerometer DFT (2 Hz) Accelerometer DFT (1 Hz)	Window size: 1 sec GPS sampling rate: 1 Hz Accelerometer sampling rate: 32 Hz Overlap: Non
[Hemminki et al., 2013]	Mean, Variance, Median, Min, Range, Interquartile range Kurtosis, Skewness RMS, Integral Double Integral Auto-correlation Mean-crossing rate FFT DC, 1, 2, 3, 4, 5, 6 Hz Spectral Energy Spectral Entropy Spectrum peak position Wavelet Entropy Wavelet Magnitude Volume, Intensity, Length Variance of peak features Peak frequency Stationary duration Stationary frequency	Window size: 1.2 sec Accelerometer sampling rate: 60 Hz and 100 Hz Overlap: 50%
[Wang et al., 2010]	Mean Standard Deviation Mean crossing rate Third quartile Sum and standard deviation of freq. Components 0-2Hz Ratio of freq. Components 0-2Hz to all frequency components Sum and standard deviation of freq. Components 2-4Hz Ratio of freq. Components 2-4Hz to all frequency components Spectrum peak positions For the vertical component Same features For the horizontal component Same features	Window size: 8 sec Accelerometer sampling rate: 35 Hz FFT over first 256 samples Overlap: Non

Table 4: Accuracy

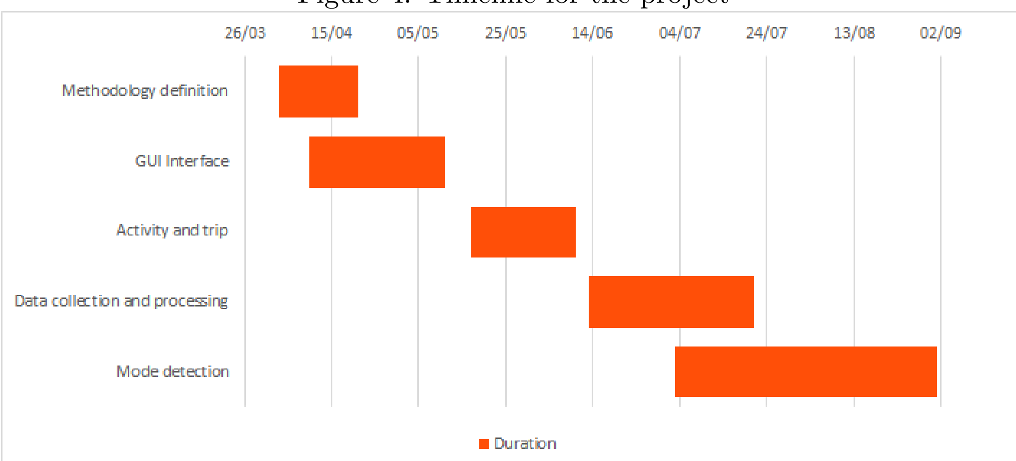
Paper	Accuracy	Comments
[Zheng et al., 2008]	Around 80%	Best results using Decision tree and change point detection algorithm
[Schüssler and Axhausen, 2009]	NA	The results were compared with a previous survey in 2005, Swiss Microcensus
[Bolbol et al., 2012]	Around 80%	Better results on acceleration cycle, bus, tube are poorly classified
[Tsui and Shalaby, 2006]	Around 90%	
[Dalumpines and Scott, 2017]	Around 90%	Bus is classified incorrectly as car
[Zong et al., 2015]	Around 90%	
[Reddy et al., 2010]	Around 90%	
[Wang et al., 2010]	Around 70%	
[Hemminki et al., 2013]	Around 80%	Difficult to separate train from metro Using only frame features accuracy drops to 60%

4 Methodology

We followed a hierarchical decomposition of the trajectory raw data, that is more suited for our research in mobility. As the example in Figure 1 shows we identified the activity points by finding points of interests (POIs). In the first iteration of our method we identified a Trip as the trajectory between two POIs, and for each Trip we will test machine learning methods to identify the transportation mode. In successive refinements we will explore methods for finding the changing point for each trip with the same transport mode.

The project was divided in the following functional activities: a first phase of problem and methodology definition, a development of a user interface for validation, the activity and trip detection, data collection and processing, and finally the application of machine learning methods for mode detection. We had a delay in the collection of data from sensors.

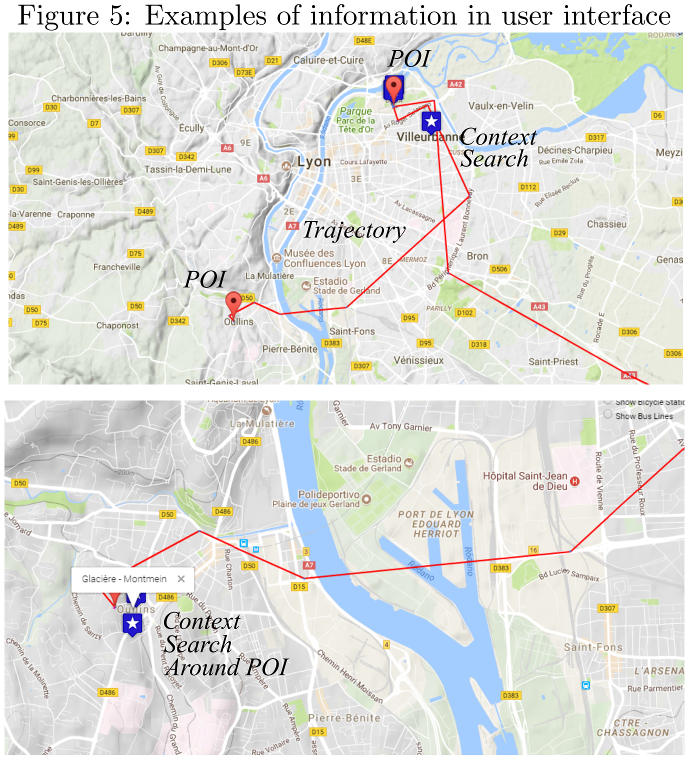
Figure 4: Timeline for the project



4.1 User Interface

In order to understand and validate the data generated we developed some visualization technique for further analysis and fine tuning of the capture application. A web based application was designed to show the GPS traces per user and per day. Many visualization methods were tested like R with R Studio Shiny and Leaflet. Finally the interface was built using Google Maps API. The interface can show the user trajectories per day, the POIs, and search for information around the POI to provide context and semantics. We can see some information in Figure 5

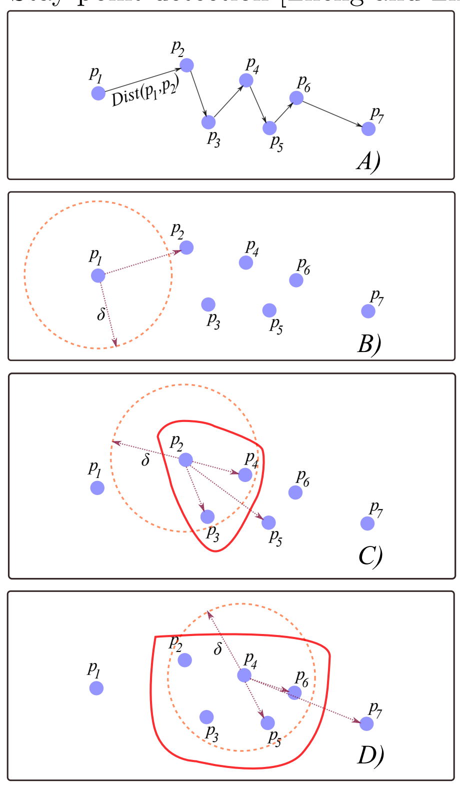
Figure 5: Examples of information in user interface



4.2 Identification of Activities, Trips and Transportation Mode

It was imperative to find a method to simplify the display of information. First it was noticed that when a user is inside a building or a place during a relatively large amount of time, some GPS captures are generated but they don't correspond to a movement. They could correspond to movement inside a shopping mall for example. We would like to group all theses captures inside a region which we can identify as an activity zone. There are some algorithms to find them and cluster the points in the vicinity, some of the methods use density based clustering like DBSCAN[3] or OPTICS[4]. Figure 6 describes the process for stay points detection as proposed by [Zheng and Zhou, 2011, p. 254]. This algorithm requires two parameters: a distance threshold δ, and a time threshold τ. For visualization purpose many combinations of δ and τ were tested and added to the GUI tool for validation. To simplify the inference model, we are going to consider at this phase of the project, that a Trip is the trajectory between two POIs.

Figure 6: Stay point detection [Zheng and Zhou, 2011]



For POIs detection we tested different values of δ and τ. It was found that a combination of 200 mts for δ and a τ 30 min gave acceptable results, allowing us to reduce the number of points inside an activity region.

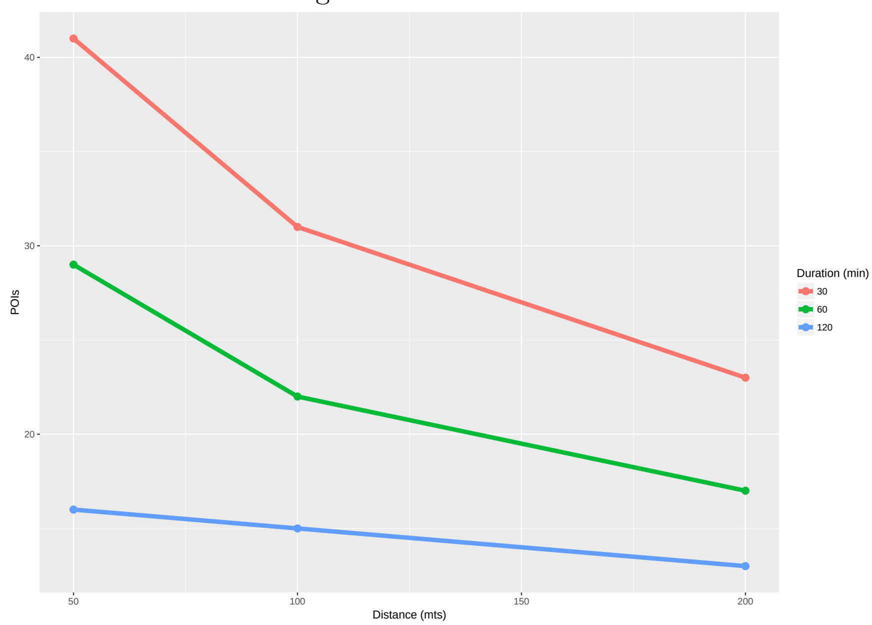
In Figures 7 and 8 we compared the number of POIs detected for different values of duration and distance threshold, and for the average number of points clustered inside the POIs detected. As expected the higher the threshold on distance δ, the lower the number of POIs detected but the higher the number of points clustered inside the POIs. Similarly the higher the duration threshold τ, the lower the number of POIs detected but the higher the points clustered inside the POI. As a visual demonstration we ran the algorithm in extreme

[3]Density-based spatial clustering of applications with noise

[4]Ordering points to identify the clustering structure

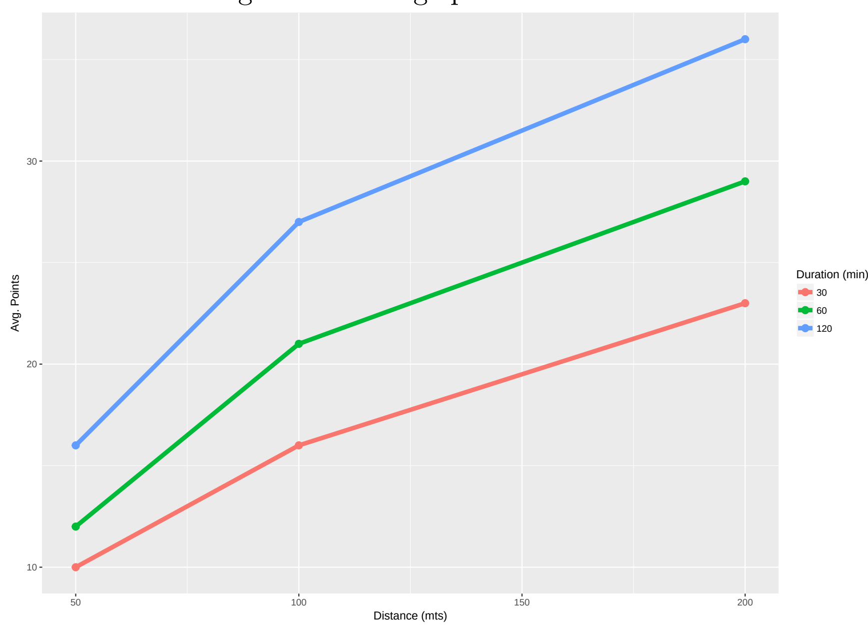
cases. For the first one are 20 mts and 30 min, and for the second one are 200 mts and 30 min. The red markers are the POIs detected. The results are shown in Figures 9 and 10, and we can appreciate that with a distance threshold of 20 mts we have lots of points around POIs that are not clustered inside them, complicating the drawing of the trajectory, while in the case of a threshold of 200 mts, all those points are clustered inside a single POI. The results go in line with research in the area in which acceptable parameter are 200 mts and 15 min [Zheng and Zhou, 2011, p. 256].

Figure 7: POIs detected



After finding a proper method for segmentation, then we can improve the Trip identification. Finally, for Mode detection we will use each Trip to classify them as one of the following types: walk, car, tramway, train, bus, and bike.

Figure 8: Average points inside POI



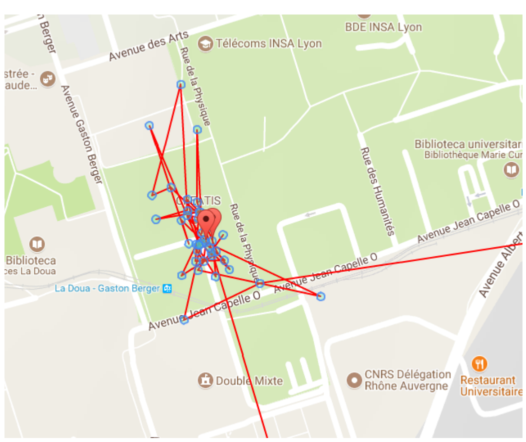


Figure 9: POIs for $\delta = 20$ mts, $\tau = 30$ min

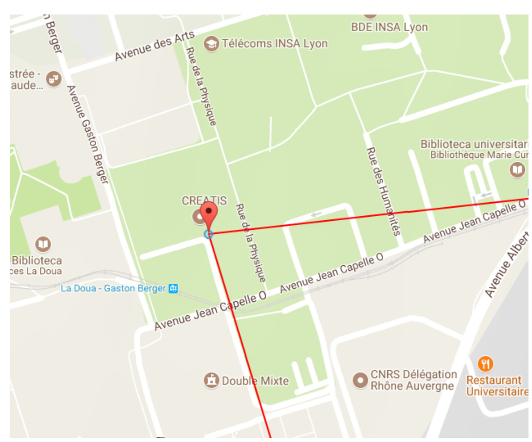
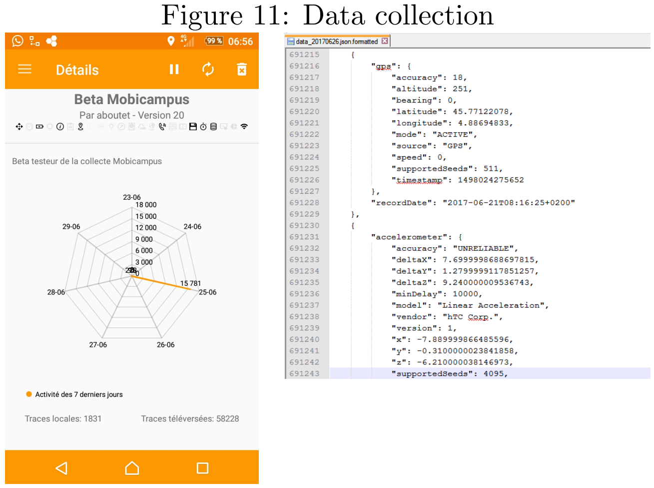


Figure 10: POIs for $\delta = 200$ mts, $\tau = 30$ min

4.3 Data collection and processing

For data generation and collection, we use an application designed for the Android operating system that captures data from many sensors inside the mobile phone, like the GPS sensor, the accelerometer sensor, network scans from Wifi networks around the device, and other sensors. Because the application requires a big set of parameters and threshold settings, we are testing plausible configurations for the final setup. After collecting data from users, it was found that we could improve the mobility traces for users adding a wifi scan to the capturing process. We are currently testing the feasibility of extracting and infer the location coordinates (latitude, longitude) given a network scan of the wifi access points surrounding a device. In Figure 11 we see the mobile phone app screen and the data generated in json format.

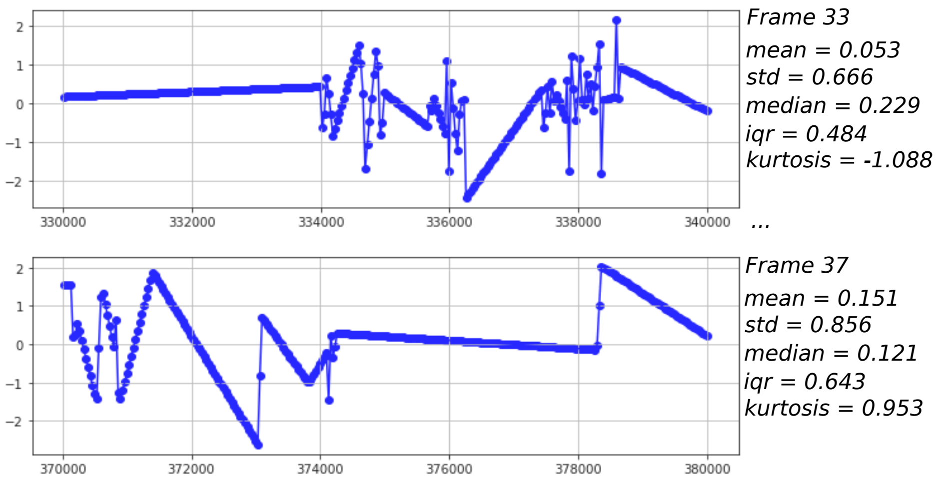
Figure 11: Data collection



4.3.1 Feature extraction

For acceleration data we followed the approaches suggested in research. We use a sliding window signal processing method, and the features extracted on a frame by frame basis are: mean, standard deviation, median, min, max, range, interquartile range, kurtosis, and skewness. All of them for each component (x, y, z) of instantaneous acceleration. The same features are extracted for each component of the derivate of instantaneous acceleration (delta). The feature extraction is parameterized by the sampling rate and the size of the window in seconds. We are not using overlapping windows. In the Figure 12 we can see the calculations of features on a sample signal on a frame by frame basis, with sampling rate of 30 Hz and window size of 10 sec.

Figure 12: Frame feature extraction



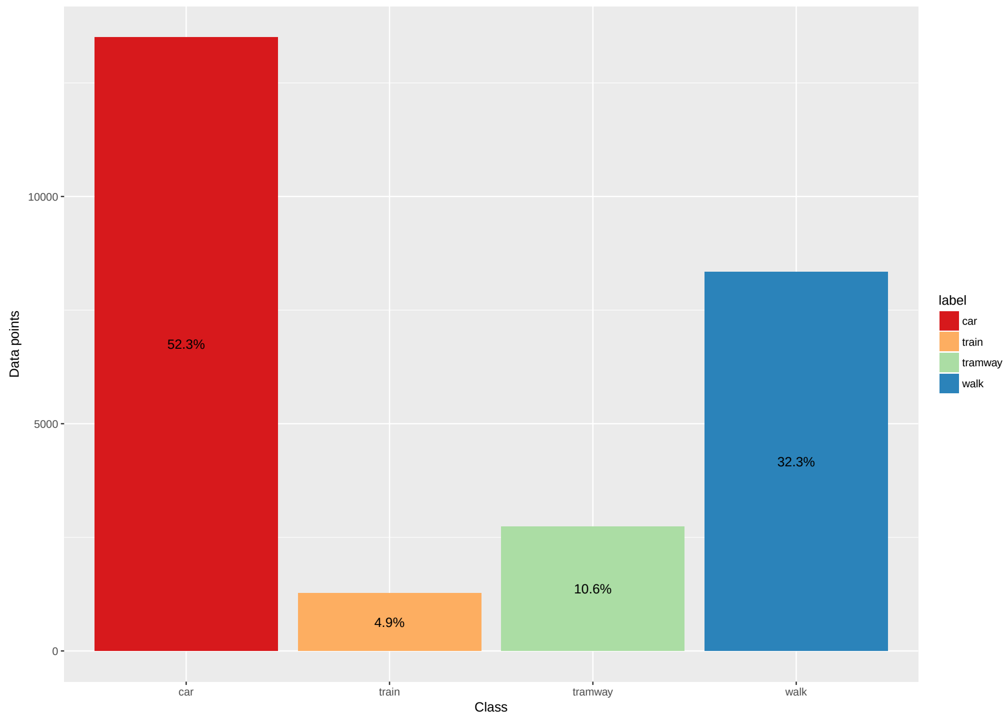
4.3.2 Classification on experimental data

We generated, collected, and labeled data from accelerometer for two devices and considering 4 classes (car, tramway, train, and walk). In Figure 13 the distribution of data points according to the classes. We prepared and additional segment labeled as 'car' for future testing.

With this raw data we computed features for the acceleration and delta component of the acceleration signals, generating data sets with 2 seconds windows and 5 seconds windows. The signals are sampled at 30 Hz. Using windows of 2 seconds we have 12342 frames for the class car, 4096 for train, 1889 for tramway, and 3164 for walk. Using windows of 5 seconds we have 4938 frames for car, 1619 for train, 757 for tramway, 1266 for walk. The total number of features extracted are 75 which include statistical features and frequency domain features. A complete list of features are shown in Table 11.

At first we tried to use a feature selection algorithm for reducing the number of features, but the results weren't conclusive. Instead we trained classifiers with two groups of features, selecting features described in the research literature. The features for group 1 are: f1_max_acc (max value of the magnitude of acceleration vector), f1_mean_acc (mean of the magnitude of acceleration vector), f1_median_acc (median of the magnitude of acceleration vector), f1_min_acc (min of the magnitude of acceleration vector), f1_std_acc (standard deviation of the magnitude of acceleration vector), f2_max_dacc (max value of delta acceleration), f2_mean_dacc (mean of delta acceleration), f2_median_dacc (median of delta acceleration), f2_min_dacc (min value of delta acceleration), f2_std_dacc (standard deviation of delta acceleration). For group 2: f1_max_acc (max value of the magnitude of acceleration vector), f1_mean_acc (mean of the magnitude of acceleration vector), f1_min_acc (min value

Figure 13: Data set captures distribution



of the magnitude of acceleration vector), f1_std_acc (standard deviation of the magnitude of acceleration vector), f3_abs_fft_1hz (Magnitude of fft of acceleration vector at at 1 Hz), f3_abs_fft_2hz (Magnitude of fft of acceleration vector at at 2 Hz), f3_abs_fft_3hz (Magnitude of fft of acceleration vector at at 3 Hz).

The results of the classifiers on test data are shown in Tables 5 to 8. It seems that using random forest and gradient boosting classifiers we have a fair average precision in the group of features 1 and using windows size of 2 seconds. Nevertheless there is a low recall in classes 'tramway' and 'train'. This can be explained by the fact that we had less data captures in this type of motorized vehicles. In a further test we labeled segments of 2 independent trips as 'car' and tested using our classifiers. After feature extraction using a 2 seconds window this test data set consists of 1271 frames. The results obtained are poor. Although the majority of frames are predicted as car, there are frames predicted as the other classes, this can be noted in low Recall values, for example for a Random Forest classifier using the features in the group 1 we got 0,45 and using Gradient Boosting we get 0,49; while using the group 2 of features we got 0,49 for Random Forest and 0,50 for Gradient Boosting (see Tables 9 and 10). The low performance can be explained because for these trips there were fragments in the trip that required many stops (stop signs, traffic lights, etc), and this requires a fine grain level of labeling, which wasn't possible will doing the data collection.

Table 5: Group of features 1, window size 2 sec

Class	**Decision tree**		**SVC rbf**		**Random forest**		**Gradient boost.**	
	Precision	Recall	Precision	Recall	Precision	Recall	Pecision	Recall
car	0,90	0,91	0,82	0,90	0,88	0,98	0,89	0,97
train	0,80	0,82	0,73	0,63	0,91	0,79	0,90	0,81
tramway	0,72	0,64	0,64	0,47	0,98	0,59	0,95	0,61
walk	0,97	0,96	0,94	0,88	0,97	0,98	0,98	0,98
avg	**0,88**	**0,88**	**0,80**	**0,81**	**0,91**	**0,91**	**0,91**	**0,91**

Table 6: Group of features 1, window size 5 sec

Class	**Decision tree**		**SVC rbf**		**Random forest**		**Gradient boost.**	
	Precision	Recall	Precision	Recall	Precision	Recall	Pecision	Recall
car	0,88	0,87	0,80	0,91	0,87	0,97	0,88	0,96
train	0,71	0,74	0,69	0,61	0,88	0,73	0,85	0,77
tramway	0,56	0,53	0,62	0,33	0,92	0,49	0,91	0,54
walk	0,96	0,97	0,95	0,85	0,95	0,98	0,97	0,98
avg	**0,83**	**0,83**	**0,78**	**0,79**	**0,89**	**0,89**	**0,89**	**0,89**

Table 7: Group of features 2, window size 2 sec

Class	**Decision tree**		**SVC rbf**		**Random forest**		**Gradient boost.**	
	Precision	Recall	Precision	Recall	Precision	Recall	Pecision	Recall
car	0,83	0,92	0,76	0,88	0,84	0,95	0,83	0,94
train	0,70	0,58	0,48	0,44	0,76	0,59	0,74	0,58
tramway	0,65	0,43	0,49	0,25	0,78	0,43	0,77	0,44
walk	0,92	0,92	0,93	0,73	0,92	0,96	0,93	0,94
avg	**0,80**	**0,81**	**0,71**	**0,72**	**0,83**	**0,83**	**0,82**	**0,83**

Table 8: Group of features 2, window size 5 sec

Class	**Decision tree**		**SVC rbf**		**Random forest**		**Gradient boost.**	
	Precision	Recall	Precision	Recall	Precision	Recall	Pecision	Recall
car	0,81	0,91	0,75	0,90	0,82	0,94	0,82	0,93
train	0,64	0,53	0,45	0,36	0,71	0,55	0,69	0,54
tramway	0,62	0,31	0,48	0,25	0,73	0,35	0,65	0,34
walk	0,90	0,94	0,96	0,71	0,91	0,94	0,91	0,94
avg	**0,78**	**0,79**	**0,70**	**0,71**	**0,80**	**0,81**	**0,79**	**0,81**

Table 9: Test on car trips (group of features 1)

Predicted	**Random forest**	**Gradient boosting**
car	576	622
train	344	317
tramway	248	190
walk	103	142

Table 10: Test on car trips (group of features 2)

Predicted	**Random forest**	**Gradient boosting**
car	624	634
train	331	304
tramway	239	251
walk	77	82

Table 11: Complete list of features

Feature	Description
f1_mean_x	Mean of acceleration component in x axis
f1_mean_y	Mean of acceleration component in y axis
f1_mean_z	Mean of acceleration component in z axis
f1_mean_acc	Mean of magnitude of acceleration vector
f1_std_x	Standard deviation of acceleration component in x axis
f1_std_y	Standard deviation of acceleration component in y axis
f1_std_z	Standard deviation of acceleration component in z axis
f1_std_acc	Standard deviation of magnitude of acceleration vector
f1_median_x	Median of acceleration component in x axis
f1_median_y	Median of acceleration component in y axis
f1_median_z	Median of acceleration component in z axis
f1_median_acc	Median of magnitude of acceleration vector
f1_min_x	Min of acceleration component in x axis
f1_min_y	Min of acceleration component in y axis
f1_min_z	Min of acceleration component in z axis
f1_min_acc	Min of magnitude of acceleration vector
f1_max_x	Max of acceleration component in x axis
f1_max_y	Max of acceleration component in y axis
f1_max_z	Max of acceleration component in z axis
f1_max_acc	Max of magnitude of acceleration vector
f1_range_x	Range of acceleration component in x axis
f1_range_y	Range of acceleration component in y axis
f1_range_z	Range of acceleration component in z axis
f1_range_acc	Range of magnitude of acceleration vector
f1_iqr_x	Interquartile range of acceleration component in x axis
f1_iqr_y	Interquartile range of acceleration component in y axis
f1_iqr_z	Interquartile range of acceleration component in z axis
f1_iqr_acc	Interquartile range of magnitude of acceleration vector
f1_kurtosis_x	Kurtosis of acceleration component in x axis
f1_kurtosis_y	Kurtosis of acceleration component in y axis
f1_kurtosis_z	Kurtosis of acceleration component in z axis
f1_kurtosis_acc	Kurtosis of magnitude of acceleration vector
f1_skew_x	Skewness of acceleration component in x axis
f1_skew_y	Skewness of acceleration component in y axis
f1_skew_z	Skewness of acceleration component in z axis
f1_skew_acc	Skewness of magnitude of acceleration vector
f2_mean_dx	Mean of delta component in x axis
f2_mean_dy	Mean of delta component in y axis
f2_mean_dz	Mean of delta component in z axis
f2_mean_dacc	Mean of magnitude of delta
f2_std_dx	Standard deviation of delta component in x axis
f2_std_dy	Standard deviation of delta component in y axis
f2_std_dz	Standard deviation of delta component in z axis
f2_std_dacc	Standard of magnitude of delta
f2_median_dx	Median of delta component in x axis
f2_median_dy	Median of delta component in y axis
f2_median_dz	Median of delta component in z axis
f2_median_dacc	Median of magnitude of delta
f2_min_dx	Min of delta component in x axis
f2_min_dy	Min of delta component in y axis
f2_min_dz	Min of delta component in z axis
f2_min_dacc	Min of magnitude of delta
f2_max_dx	Max of delta component in x axis
f2_max_dy	Max of delta component in y axis
f2_max_dz	Max of delta component in z axis
f2_max_dacc	Max of magnitude of delta
f2_range_dx	Range of delta component in x axis
f2_range_dy	Range of delta component in y axis
f2_range_dz	Range of delta component in z axis
f2_range_dacc	Range of magnitude of delta
f2_iqr_dx	Interquartile range of delta component in x axis
f2_iqr_dy	Interquartile range of delta component in y axis
f2_iqr_dz	Interquartile range of delta component in z axis
f2_iqr_dacc	Interquartile of magnitude of delta
f2_kurtosis_dx	Kurtosis of delta component in x axis
f2_kurtosis_dy	Kurtosis of delta component in y axis
f2_kurtosis_dz	Kurtosis of delta component in z axis
f2_kurtosis_dacc	Kurtosis of magnitude of delta
f2_skew_dx	Skewness of delta component in x axis
f2_skew_dy	Skewness of delta component in y axis
f2_skew_dz	Skewness of delta component in z axis
f2_skew_dacc	Skewness of magnitude of delta
f3_abs_fft_1hz	Magnitude of fft of magnitude of acceleration vector at at 1 Hz
f3_abs_fft_2hz	Magnitude of fft of magnitude of acceleration vector at at 2 Hz
f3_abs_fft_3hz	Magnitude of fft of magnitude of acceleration vector at 3 Hz

4.3.3 Geolife data set

In order to test the accuracy of the methods in the literature and have a baseline, we needed labeled data sets. At the moment the only public available data set is the geolife data set. It consists of logged GPS trajectories for a period of 4 years. Most of the trajectories are logged every 1 to 5 seconds [Zheng et al., 2011]. This data set is used in [Zheng et al., 2008]. Each GPS record contains Latitude, Longitude, Altitude in feet, and the Date. There are separate files for each day and for each user (total of 178 users). The transportation mode labels are stored in separate files per user and they consists of text files with the Start Time, End Time, and the Label describing the mode of transportation. We built a data processing work flow to get a more useful representation of the labeled data set and processed only 50% of the original data set. Finally we got more than 3 Million GPS labeled records. Further processing had to be applied to the raw data to divide the trajectories further into trips if the time between 2 consecutive records is higher than a threshold ([Zheng et al., 2008] uses a 20 minutes threshold). The speed is not provided in each of the GPS records, therefore it had to be calculated using a great-circles distance between two consecutive coordinates, and the difference in time between the them. A similar approach had to be followed to estimate the acceleration between consecutive locations dividing the speed by the time difference.

In order to test some learning methods, we tried to apply the algorithm for change point detection and segmentation. A brief description of the steps is provided here [Zheng et al., 2008, p 5]:

1. Step 1: Records with speed less that 1.8 m/s and acceleration less than 0.6 m/s^2 are considered as possible Walk Points. The rest are considered possible Non-Walk Points.

2. Step 2: If the length of a segment made of consecutive Walk Point or Non-Walk Points are less than a threshold, merge them into the previous segment.

3. Step 3: If the length of a segment exceeds 50 mts, the segment is regarded as Certain. Otherwise is considered Uncertain. If the number of consecutive uncertain segments are more 3, these Uncertain segments are merge into a Non-Walk segment.

4. The start point and end point of each Walk segment are the potential change points.

At the moment we tried to applied the rules described but the threshold for Step 2 is not found in the literature. We tried to use a 2 mts threshold without conclusive results.

At this phase we only tested machine learning methods on a segment by segment basis knowing that the beginning and end of a labeled section of each trajectory demarks the limits of each segment, without using the change point detection algorithm. Because [Zheng et al., 2008] uses only 4 classes, we discarded irrelevant ones like boat, or airplane. Finally, we divided the trajectories into segments and calculated 12 features. The final data set consists of 5820 segments. Using Weka 3.8.1 [Hall et al., 2009] we tested three methods using a split 80/20 for training and testing with the following methods: J48 Decision Tree, Bayes Net, and Naives Bayes. The Decision Tree method was the best with an accuracy of 83.4%. The average recall is similar to [Zheng et al., 2008] when they use the change point detection method, but the precision differs from the study (40.6% in [Zheng et al., 2008] versus 83.3% in our case). The results are not yet conclusive because we have to validate the change point detection algorithm.

Figure 14: Confusion matrix

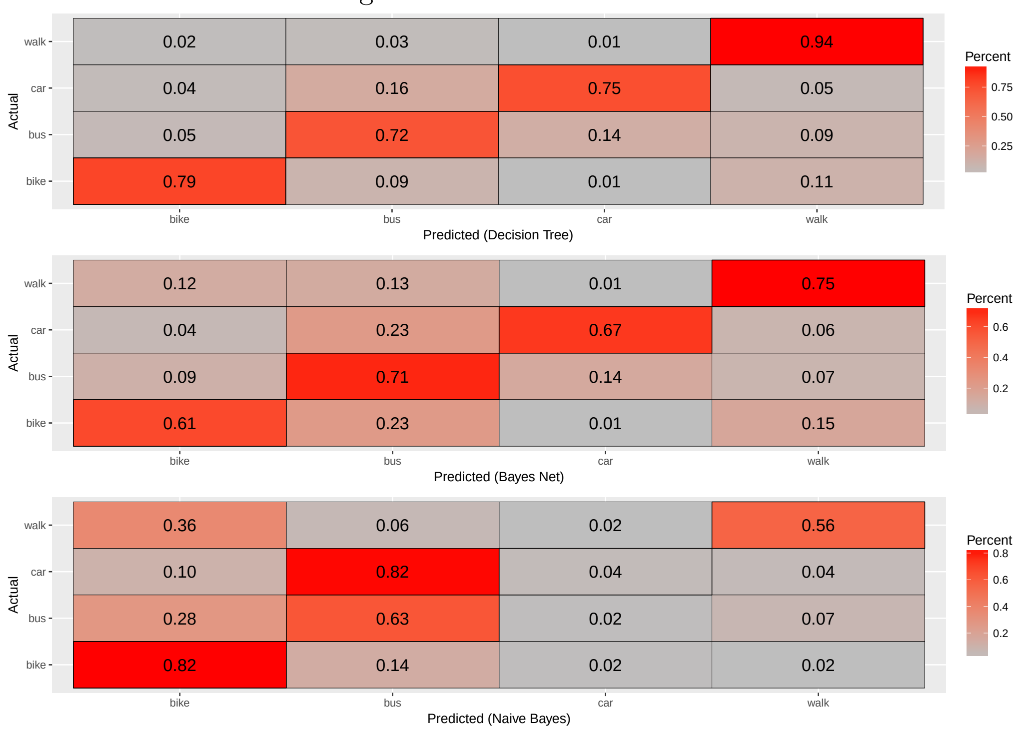
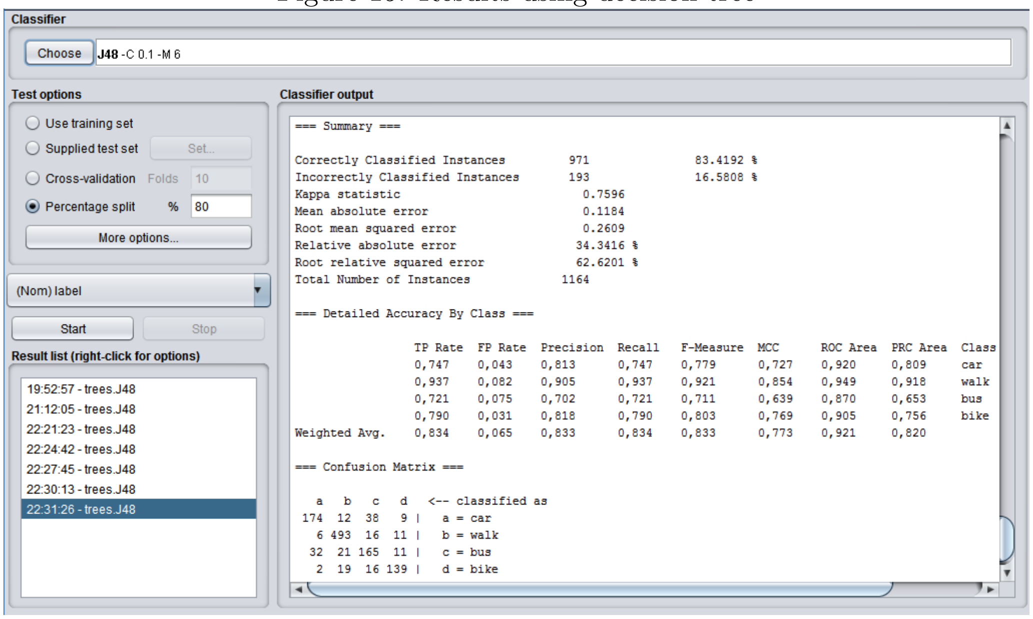


Figure 15: Results using decision tree



5 Conclusions and Perspectives

For GPS data, our approach to reduce the size of the problem by finding first the activities through a POIs detection algorithm could make easier the detection in the next phases because our approach reduces the sizes of the trajectories.

As seen in the research literature, the use of a GPS data based approaches differs from the use of acceleration data ([Reddy et al., 2010] uses GPS and accelerometer data), mostly due to the nature of the capture rates and the semantics of data. Geospatial information from GPS is captured at slower rates than acceleration, and also for GPS data the concept of a geographical change point seems to fit naturally, while in acceleration data not. Also acceleration data can be treated as an instance of a signal processing problem (involving frequency domain analysis, wavelets, etc). We have to analyse and select the more appropriate sensor data to our problem, or combine the results of different classifiers.

With respect to baselines, there remains few things to try. The first is to try to replicate the algorithm used in [Zheng et al., 2008] for detecting the change points and apply the machine learning methods described in the paper. The second is to apply other machine learning methods on raw or pre-processed GPS data to find the changing points themselves. Interesting ideas to try: the use of recurrent neural networks for sequential data.

Regarding acceleration, using the labeled data sets we built, at first it seems we can discriminate at least some form of motorized vehicle from walking. Nevertheless our final test on two 'car' trips have shown poor results, though the predictions were correct most of the time. As pointed out by the literature, acceleration is more suitable for a kind of 'real time' prediction of the transport mode [Reddy et al., 2010]; therefore we have to include in the algorithm some threshold based mechanism to activate the classifier. If the threshold (based on variance, for example) is not reached, the prediction algorithm should output 'no movement'. We also can consider a probabilistic model (for example HMM) [Reddy et al., 2010], on top of an instance based classifier.

Future works should consider adding speed, but this parameter is obtained from the gps sensor, with different capture timelines, therefore there must be some data preprocessing to merge and combine the two types of signals. Also, as pointed out by the literature, we should consider the process of segmentation to find out the points in the trajectory in which the user changes the transportation mode.

References

[Bolbol et al., 2012] Bolbol, A., Cheng, T., Tsapakis, I., and Haworth, J. (2012). Inferring hybrid transportation modes from sparse {GPS} data using a moving window {SVM} classification. *Computers, Environment and Urban Systems*, 36(6):526 – 537. Special Issue: Advances in Geocomputation.

[Dalumpines and Scott, 2017] Dalumpines, R. and Scott, D. M. (2017). Making mode detection transferable: extracting activity and travel episodes from gps data using the multinomial logit model and python. *Transportation Planning and Technology*, 40(5):523–539.

[ESRI, 2003] ESRI (2003). Mean sea level, gps, and the geoid. [Online; accessed 01-June-2017].

[Gong et al., 2014] Gong, L., Morikawa, T., Yamamoto, T., and Sato, H. (2014). Deriving personal trip data from gps data: A literature review on the existing methodologies. *Procedia - Social and Behavioral Sciences*, 138:557 – 565. The 9th International Conference on Traffic and Transportation Studies (ICTTS 2014).

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

[Hemminki et al., 2013] Hemminki, S., Nurmi, P., and Tarkoma, S. (2013). Accelerometer-based transportation mode detection on smartphones. In Petrioli, C., Cox, L. P., and Whitehouse, K., editors, *SenSys*, pages 13:1–13:14. ACM.

[IMU, 2016] IMU (2016). Mobicampus-udl - articuler recueil de donnes d'enqutes web et suivi de traces de tlphonie mobile pour comprendre et manager la mobilit des universitaires du campus de l'universit de lyon (2016).

[Mizell, 2003] Mizell, D. (2003). Using gravity to estimate accelerometer orientation. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ISWC '03, pages 252–, Washington, DC, USA. IEEE Computer Society.

[Reddy et al., 2010] Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., and Srivastava, M. (2010). Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27.

[Renso et al., 2013] Renso, D. C., Spaccapietra, D. S., and Zimnyi, D. E. (2013). *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press, New York, NY, USA.

[Schüssler and Axhausen, 2009] Schüssler, N. and Axhausen, K. W. (2009). Processing GPS raw data without additional information. *Transportation Research Record*, 2105:28–36.

[Tsui and Shalaby, 2006] Tsui, S. Y. A. and Shalaby, A. S. (2006). Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems. *Transportation Research Record: Journal of the Transportation Research Board*, (1972):38–45.

[Wang et al., 2010] Wang, S., Chen, C., and Ma, J. (2010). Accelerometer based transportation mode recognition on mobile phones. In *2010 Asia-Pacific Conference on Wearable Computing Systems*, pages 44–46.

[Wikipedia, 2017a] Wikipedia (2017a). Accelerometer. [Online; accessed 01-June-2017].

[Wikipedia, 2017b] Wikipedia (2017b). Latitude. [Online; accessed 01-June-2017].

[Wikipedia, 2017c] Wikipedia (2017c). Longitude. [Online; accessed 01-June-2017].

[Zheng et al., 2011] Zheng, Y., Fu, H., Xie, X., Ma, W.-Y., and Li, Q. (2011). *Geolife GPS trajectory dataset - User Guide.*

[Zheng et al., 2008] Zheng, Y., Liu, L., Wang, L., and Xie, X. (2008). Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 247–256, New York, NY, USA. ACM.

[Zheng and Zhou, 2011] Zheng, Y. and Zhou, X. (2011). *Computing with spatial trajectories.* Springer Science & Business Media.

[Zong et al., 2015] Zong, F., Bai, Y., Wang, X., Yuan, Y., and He, Y. (2015). Identifying travel mode with gps data using support vector machines and genetic algorithm. *Information*, 6(2):212–227.