Faculdade de Engenharia da Universidade do Porto

Extracting Mobility-Relevant Information from Crowdsourced GPS Data

João Pedro de Jesus Vieira Pereira

Master in Electrical and Computers Engineering
Telecommunications, Electronics and Computers

Supervisor: Ana Cristina Costa Aguiar (PhD)

27th June 2016

Abstract

Rural exodus leads to a continuous growing flux of people in center of big cities that brought several challenging situations in terms of urban mobility. The excess of vehicles in circulation created necessity of adapting cities and its transport infrastructures to face the flowing crowds creating new opportunities to study the behavior of the ever-changing patterns of individual and community mobility.

In order to achieve a complete overview of a whole urban area it is needed to gather information from as many individuals as possible, one of the easiest and more complete ways of doing so is using smartphones' sensors that can retrieve contextual geographic information, is important to consider that smartphones are everywhere nowadays, therefore, it is possible to take advantage of this information to make it useful to the community.

The assumption is that every GPS trace produces unique information and has relevant characteristics for analyzing the individual and its irreplaceable contribution to the system. When analyzing mobility patterns two essential characteristics have to be analyzed, the origin and destination of each commute and the transportation mode used. Several approaches to classify transportation mode have been developed through the years.

One of the main contributions of this thesis lie in joining several concepts as: trip segmentation, clustering trajectory points into stay points and transportation mode detection. Another key contribution of this thesis is a new methodology to compare transportation mode speed profiles to classify a trip.

This thesis analysis is focused on the urban region of Porto and data was collected using SenseMyFEUP application.

The final solution to infer transportation mode is based on the following procedures: 1) Data filter for speed outlier detection and to filter ground-truth data (labelled trips) based on sanity analysis, 2) Trip segmentation based on stationary moments, 3) Clustering trajectory points into Stay Points using segmentation information to divide in categories of stationary and moving Stay Points and 4) Transportation mode detection.

Trip segmentation and respective clustering into stay point achieve a reduction to 1,65% of the original number of entries from the GPS trace.

Regarding transportation mode detection, in this thesis a multi-stage solution is developed having a modular component that can incorporate two methods to classify transportation mode based on speed profile. The algorithm to infer transportation mode uses speed distribution comparison and closeness to public transportation stops. When comparing speed distribution one of the methods used, Decision Tree, is recurrent in area of transportation mode inference. The other method is used for the first time in this application

field and relies on boxplot comparison between transportation mode speed profile and trip being evaluated.

Using GPS logs from 239 people over a period of three months to evaluate proposed approach results achieved report an overall accuracy of 92,60%, an overall precision of 81,56% and an overall recall of 90,86%.

Resumo

O êxodo rural levou a um contínuo aumento do fluxo de pessoas para as grandes cidades criando várias situações problemáticas na área da mobilidade urbana. O excesso de veículos em circulação e a progressiva mutação dos padrões de mobilidade das comunidades criam uma constante necessidade de melhoramento das infraestruturas e da adaptação da rede de transportes para responder às necessidades dos que nelas habitam e por lá passam.

Para se obter uma vista geral completa de uma área urbana é necessário reunir informação do maior número de indivíduos possível, uma das formas mais fáceis e completas de o fazer é recorrendo aos sensores de *smartphones* capazes de recolher informação geográfica. É importante ter em conta que nos dias de hoje os *smartphones* estão em todo o lado. Assim, é possível, tirar partido desta informação tornando-a útil para a comunidade.

É assumido que cada movimentação produz informação única e com características relevantes para analisar o perfil individual de cada um e o seu insubstituível contributo para um todo, a comunidade.

Na análise de padrões de mobilidade existem duas características essenciais que devem ser analisadas, a origem e destino de cada viagem e o modo de transporte utilizado. Várias metodologias para classificar automaticamente o meio de transporte têm vindo a ser desenvolvidas ao longo dos anos.

O contributo desta dissertação foca-se essencialmente na junção de técnicas e conceitos que, apesar de existentes, nunca foram utilizados como um todo numa solução única e transversal à análise de padrões de mobilidade. Conceitos como segmentação de viagem, agrupamento de pontos GPS em *Stay Points* e deteção do meio de transporte. Outro contributo importante desta tese é uma nova metodologia para classificar meios de transporte através da análise da distribuição das velocidades numa viagem.

A análise desta dissertação foca-se na região urbana do Porto e a informação foi recolhida utilizando a aplicação SenseMyFEUP.

O produto final relativo à detecção de modo alicerça-se na combinação das seguintes soluções para resolver problemas individuais: 1) Filtragem de dados através da deteção de valores atípicos de velocidade, 2) Segmentação de viagem, sendo as quebras efetuadas em momentos sem movimento 3) Agrupamento de pontos GPS em *Stay Points* com base na informação obtida na segmentação sobre pontos em movimento e pontos estacionários e 4) Deteção de meio de transporte.

A segmentação de viagem e agrupamento dos respetivos pontos em *Stay Points* permitiu uma redução para 1,65% dos dados originais relativos a viagens.

Relativamente à deteção de meio de transporte, nesta dissertação foi desenvolvida uma solução multinível com uma componente modular. Esta componente modular pode incorporar

um de dois métodos para classificar a viagem baseando-se na sua distribuição de velocidades. O algoritmo para classificação de meio de transporte utiliza comparação da distribuição de velocidades e proximidade às paragens dos transportes públicos para inferência de meio de transporte. Um dos métodos, uma árvore de decisão, é de recorrente utilização na área da inferência de meio de transporte através da análise das velocidades. O outro método é proposto pela primeira vez aplicado a esta área de estudo e baseia-se numa comparação de *boxplots* entre a viagem a ser analisada e o perfil de velocidades apreendido relativo a cada meio de transporte.

Utilizando viagens de 239 pessoas recolhidas durante um período de três meses para a solução proposta obtiveram-se resultados médios de 92,60% de exatidão, uma precisão de 81,56% e um *recall* de 90,86%.

"Building more roads to prevent congestion is like a fat man loosening his belt to prevent obesity"

Lewis Mumford

List of Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Symbols

BN	Bayesian Net
CRF	Conditional Random Fields
DBMS	Database Management System
DBSCAN	Density Based Spatial Clustering of Applications with Noise
DJ-Cluster	Density-Joinable Cluster
DT	Decision Tree
GIS	Geographic Information System
GPS	Global Positioning System
MAD	Median Absolute Deviation
ML	Multilayer Perceptron
NB	Naïve Bayes
RF	Random Forest
STCP	Sociedade de Transportes Colectivos do Porto
ST-DBSCAN	Spatial-Temporal DBSCAN
SVM	Support Vector Machines

Chapter 1

Introduction

Rural exodus leads to a continuous growing flux of people in center of big cities that brings several challenging situations in terms of urban mobility. The excess of vehicles in circulation created necessity of adapting cities and its transport infrastructures to face the flowing crowds creating new opportunities to study the behavior of the ever-changing patterns of individual and community mobility.

In ubiquitous and context aware computing, understanding mobility of an individual from sensor data is an important area of research.

Participatory sensing received a big interest in scientific community that saw an opportunity in creating networks of individuals that can gather and share local knowledge about a specific region or street. This specific knowledge can then be joined in an urban sensing system, resulting in a complete overview of a whole urban area.

Analyzing urban mobility patterns allows retrieving information important to adapt existing infrastructures to the changing urban trends with expanding dimension and increasing complexity.

Several subjects might be on focus when attempting to improve the system, as is the case of optimizing networks of public transportation, create attractiveness and automatically suggest sharing private car between persons with similar routines or even make available live traffic information. One of the most representative ways of describing mobility in an urban area is through origin-destination matrices. Origin-destination matrices are a fundamental source of information for traffic control and transport planning.

In order to make possible the automatic production of this matrices, it is the needed to improve data gathering and processing. One of the easiest and more complete ways of gathering this information is using smartphones' sensors that can retrieve contextual geographic information. It is important to consider that smartphones are everywhere nowadays, therefore, it is possible to take advantage of this information to make it useful to the community.

To classify transportation mode several approaches have been developed through the years. From simpler approaches analyzing just trip speed profile that misses situations like when a user changes transportation mode during a trip to more complex developments in the

area assuming that a trip should first be segmented using stationary moments. These stationary moments being possible points of transportation mode change and each segment being evaluated alone using several heuristics like speed, acceleration, head change rate and contributing to the whole trip inference. More advanced known developments regarding transportation mode culminate in adding Geographic Information System (GIS) layers to the analysis.

As retrieving origin and destination from GPS traces is a simple task, this thesis main goal aims to automatically classify transportation modes including being stationary, walking, riding a bicycle, driving, taking a bus and taking a metro. The classification is not done directly using raw GPS logs but taking advantage of movement analysis for trip segmentation and for clustering. Applying segmentation and clustering allows reducing raw GPS traces to mobility-relevant information which translates in removing non-essential information to infer transportation mode from the analyzed data.

One of the main contributions of this thesis lie in the use of concepts such as trip segmentation, clustering trajectory points into stay points and transportation mode detection. Another key contribution is a new methodology to compare transportation mode speed profiles to classify the trip accordingly.

1.1 Motivation

Extracting mobility-relevant information from crowdsourced GPS data raises several problems.

Analyzing urban mobility patterns allows the creation of solutions to improve quality of life of a community. Nowadays people tend to spend a significant part of their life commuting either from home to work or to leisure activities.

With a good analysis of the mobility patterns, models can be designed to adapt infrastructures to fulfill the needs of an urban area, turning GPS data into concrete information useful to the individual and to the community.

The increasing ubiquity of GPS acquisition technologies allows building informative, yet unobtrusive ways of gathering data, leading to the collection of large spatiotemporal datasets, an opportunity of discovering knowledge about mobility patterns and recognition of everyday activities.

The concern with the privacy poses a critical challenge creating algorithms that are capable of analyzing patterns of mobility. Matching several traces of an individual cannot be directly done, the same problem occurs when extracting statistical information about the urban area when the subject is the movement of individuals.

Mobility patterns are still captured with outdated processes like surveying the population and compiling that information, a process that gives no detail of what really happens.

One of the most representative ways of describing mobility in an urban area is through origin-destination matrices. Origin-destination matrices are a fundamental source of information for traffic control and transport planning.

This thesis aims to fill the gap existent providing a process to achieve a greater detail when analyzing mobility patterns of an urban area.

In order to do so, algorithms to identify the origin and destination of each trip and to identify the transportation mode used have to be developed and put to use.

1.2 Goals

The primary outcomes of this dissertation comprehends the following aimed contributions to the body of knowledge in extracting mobility-relevant information:

1. **Create a clean dataset from a raw crowdsourced dataset.** How to collect data from several sources in order to simulate a real-world environment? How to identify and correct bad data? What are the characteristics that single subsets of data may illustrate that the global dataset cannot?
2. **Devise a data based algorithm to do offline mode detection.** Which information will we need? How can we cluster this information maintaining data meaningful for future information extraction? Which are the underlining necessities? Which features should be used in transportation mode detection? The outcome of this development can be found in Chapter 5.
3. **Evaluate mode detection algorithm using real-world data.** What are the specific characteristics of each transportation mode? Which is the transportation mode used in each commute? What is the distribution of the transportation modes in the studied area? How public transportation network is responding to the needs of users? In transportation planning process answers to this questions are crucial and with this dissertation it is desired to provide a tool to make this analysis easier, faster and with permanently updated data.
4. **Create an infrastructure for extracting origin-destination matrix data from real GPS traces.** Which are the underlining necessities to create origin-destination matrices?

1.3 Structure

The remaining of this dissertation is organized with the following overall structure.

- Chapter 2, “Literature Review” (p. 5), describes the state of the art, and provides and reviews related work in areas such as clustering location data, definition of locations, trip pattern analysis and detection of transportation mode.
- Chapter 3, “Problem, Technologies and Collection Tool” (p. 13), describes the problem and outlines the technologies involved in the development of this thesis system and provides an overview of the tool used to collect data.
- Chapter 4, “Data Collection and Dataset” (p. 17), presents the process of gathering data. Describes data used for training and to validate developed methods for classifying transportation mode. Additionally, is presented a detailed analysis about collected data.

- Chapter 5, “Mode Detection of GPS Traces” (p. 25), provides an insight on the implementation of the algorithms developed during this thesis.
- Chapter 6, “Results” (p. 41), presents a stance on the results obtained and it is done a comparison with the results achieved by other authors.
- Chapter 7, “Conclusion” (p. 45), draws the main conclusions of this dissertation and offers an outlook on future work. Also, highlights the contributions of this thesis, describing the novelty of developed features.

Chapter 2

Literature Review

In this chapter an analysis of the state-of-art is done presenting an overview on what is most recent and representative for the study of extracting mobility-relevant information from crowdsourced GPS data.

In search for solutions on how to reduce and extract meaning from crowdsourced raw data produced by GPS sensors, researchers were pushed to create methods and strategies to improve several subjects of study.

The following main themes need to be focus to address the problem:

1. Clustering information;
2. Identification and definition of locations;
3. Reducing location trace information;
4. Trip segmentation;
5. Transportation mode detection.

Understanding these themes is a starting point for the development of methods and algorithms to detect transportation mode from a GPS trace.

2.1 Clustering

Clustering algorithms are designed to form groups such that the members of each group are more similar compared to non-group members, similarity might be based in several characteristics. When the subject of analysis is a group of GPS points, the main characteristics that might produce interesting similarity and the ones analyzed in this thesis and in literature are spatial and temporal components of each point.

[1] introduces Density Based Spatial Clustering of Applications with Noise (DBSCAN), a spatial density-based clustering algorithm that was created to run on top of large data sets giving origin to clusters of arbitrary shapes and size. DBSCAN needs to parameters to run *Eps* and *MinPts*. The first represents the maximum distance between two points to be considered neighbors and the seconds the minimum number of neighbor points to originate a cluster. In DBSCAN clusters are defined as the maximum density-reachable points, the density associated

to a point is obtained by the number of points in a region of specified radius around the point. Clusters are constructed when having a density superior to a specified threshold. As a density-based clustering algorithm, DBSCAN does not consider time domain. DBSCAN defines clusters of high-density reachable points and can find clusters of arbitrary shape. [1] provides an example of results from DBSCAN that can be seen in Figure 1.

Figure 1 - Clustering with DBSCAN [1]

[2] introduces changes in DBSCAN giving origin to a new algorithm called Spatial-Temporal Density Based Clustering of Applications with Noise (ST-DBSCAN) with ability to cluster information based on its non-spatial, spatial and temporal attributes. ST-DBSCAN allows the clustering points with temporal similarity using distance metrics not only for geographic coordinates but also to temporal distance. This way, a point belongs to a cluster if it is inside the spatial and temporal thresholds defined.

In [3] and [4] compare two algorithms for extracting meaningful places from GPS traces. Final results showed superiority of accuracy on DJ-Cluster (Density-Joinable Cluster) against K-means algorithm. K-means algorithm creates K groups from the set of points so that the members of each group are more similar. This similarity is calculated having in account the distance between them.

DJ-Cluster calculates the neighborhood for each point. A neighborhood is formed by points within a certain distance *Eps* and to form a valid neighborhood it must have at least *MinPts* points within *Eps* radius. When these conditions are not fulfilled the point is labeled as noise and discarded.

Density based neighborhood of a point (p):

$$N(p) = \{q \in S \mid dist(p,q) \leq Eps\}, N(p) \geq MinPts$$

In prior Equation, *S* represents the set of all points, *q* represents any points, *Eps* represents circle radius around p and *MinPts* is the minimum number of points required within the circle to form a new cluster. If new calculated cluster overlaps existing clusters they are merged.

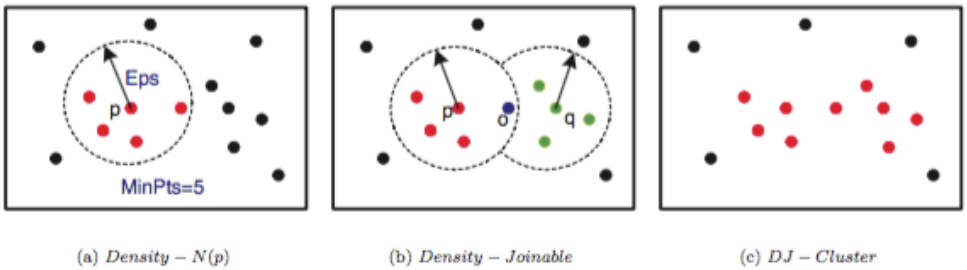


Figure 2 - Density-based join concepts [3]

Figure 2 illustrates three density based concepts: (a) The Density based neighborhood (b) N(p) density joinable to N(q) and (c) illustrates the final clusters in red.

2.2 Identification and Definition of Locations

In [5] relations between spatial movements and social context are explored. To capture this relation, the authors, use definitions like geo-location and geo-community. Concepts that, respectively, depend on time spent by an individual or a group of individuals in a place. Their analysis starts with time spent in a location, these locations can be categorized as transit-locations or main destinations depending on the amount of time spent there and also if it is common to several users or to a single user only. Coordinating these two parameters will provide the difference between transit locations (less time, single user) and main destinations (more time, community).

The problem of discovering places that matters to a person daily life and routine is addressed in [3]. Places visited by a user are labeled regarding their importance and frequency using a spatial clustering algorithm and a classifier.

[6] defines a place using moments when GPS signal is lost and later resumed. Therefore, places are buildings and never spaces in the open-air. To complement the information a time threshold and a distance threshold is used for clustering several points that can lead to a place.

2.3 Reducing location trace information

[7] define GPS trace as a collection of GPS points P = {p_1, p_2, … ,p_n}. Each GPS point $p_i \in P$ consists of latitude, longitude and timestamp information. A sequence of this points is called a GPS trajectory and its illustration is shown in Figure 3.

Stay Point is defined as a geographic region where a user stayed during a certain time interval, this is determined using time threshold and a distance threshold to delimit the geographic region. The authors developed a cluster based approach reviewing K-means, time, and density based clustering.

[5] define stay location *sl* as a set of GPS coordinates such that:

$$sl = \{p_m, p_{m+1}, \dots, p_n\} \; \forall \; m < i \leq n, \|p_m - p_i\| \leq D$$

sl does not need to be a sequence in time and D represents the distance threshold between points. Stay location is a cluster of points based on their proximity. [6] complemented the concept of stay locations giving importance to time spent there.

In [8] stay point stands for a geographic region with semantic meaning where an user stayed for a while and the authors created two different scenarios that give origin to a stay point. [8] in Figure 3 illustrate two scenarios. First scenario (illustrated as stay point 1) occurs when a user remains for a time period in a place exceeding a given time threshold, stay point is generated only from point P3 having then the same information of P3. The other scenario generates a virtual location characterized by clustering points {P5, P6, P7, P8}.

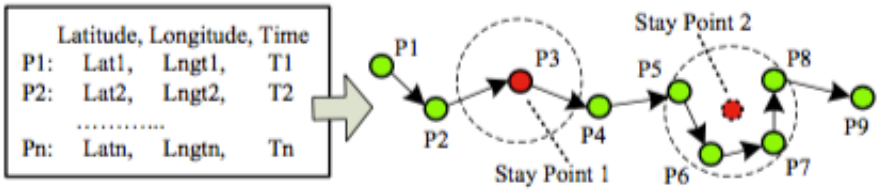


Figure 3 - GPS Logs and Stay Points [8]

To achieve these results [8] developed an algorithm to detect Stay Points. This algorithm receives the GPS trace P and two thresholds to limit the region and timespan of points to cluster. GPS points that fall inside these limits give origin to a new cluster having it geometric center in the arithmetic mean coordinate (centroid) of all points. Temporal information equals average time of occurrence of aggregated points.

[9] considers Stay Point as all points that are within a radius threshold of 125m from the reference point and time variance inferior to a threshold of 120s; the author bases his algorithm in the one developed by [8]. After detecting stay points in a GPS trace, [9] verifies if the first and last point of the trace generated stay points. If not, it is created a stay point with characteristics correspondent to start and/or end point. The author found issues when a user moved through a tunnel losing GPS signal, the system would detect a stay point in the entrance of the tunnel due to the time and distance between the entrance and the exit of the tunnel being superior to time and distance thresholds. To solve this tunnel problem, algorithm discards points that, between them, have a distance superior to 2 km and an interval superior to 2 minutes.

2.4 Trip Segmentation

[10] point out that most studies tend to presume a single transportation mode being used in a trip. This assumption may lead to wrong classification as very often, during a commute, citizens use more than one transportation mode. [10] use stops to segment trips, these stops are identified when speed is no higher than 2km/h during a 12 seconds interval.

Either way, several older studies explore the possibility of a trip having more than one transportation mode. Methods to locate those transportation modes in a trip are presented for long in the Literature, e.g. [11], [12], [13] base their segmentation on the assumption that a walking or stationary segment is required for mode change, change point-based segmentation. Figure 4 illustrates an example of a GPS trajectory with walking segment, non-waling segment and change point.

These studies look at GPS trace and try to identify stationary moments and walking points based on speed, time and distance. Points falling in those categories are candidate points to a potential transportation mode change.

[12] use a threshold of 2,8m/s for walking speed and a minimum duration of 60 seconds to identify a walking segment; the authors identify a potential mode transfer points when an end of walk, start of walk, or end of gap is detected. Gap is defined as signal lost for at least 120 seconds.

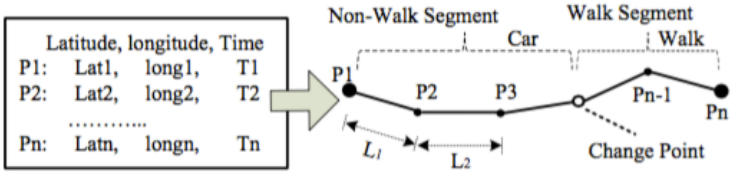


Figure 4 - GPS log, segment and change point [11]

[13] segment trips by identifying stationary segments (stops). To detect stationary segments two criteria are verified:

- no change in location (not moving at least 5 meters in 5 seconds);
- speed inferior to 0.5m/s for 5 seconds, and heading change superior to 100 decimal degrees.

The author goes further in development of rules to segment a GPS trace in subsets trying to avoid segmenting a trace where there is no change in travel mode. If two consecutive segments have same detected travel mode, they are merged. Merge procedure has to obey some rules to avoid errors and also to protect from situations when signal lost:

- travel-mode should have at least 120 seconds;
- stop duration between two different travel-modes longer than 120 seconds;
- stop duration during same segment has to be less than 120 seconds.

2.5 Detecting Transportation Mode

While several authors do a deep study in classifying a trip assuming a single transportation mode was used, this literature review focuses on researches that have broader studies preparing its approaches to situations where more than one transportation mode is used. Authors as [11], [12], [13] identify these transitions between transportation modes considering that rare are the cases when a transition between travel modes is not accompanied by a stationary segment and a walking segment.

[13], after segmenting GPS trace, employ a multi-step method to detect transportation mode: in first step the traces correspondent to pedestrian and bicycle travel modes are identified and is assumed that the rest are motor-based segments with a reported certainty of 94%; in second step motor-based segments are linked up to create sub-traces and motor-based travel modes are classified as being car, bus, tram or train. In second step classification the author used Support Vector Machines (SVM) based on 11 parameters (means and standard deviations of average and maximum speed, acceleration, average acceleration, travel time and stop rate) to classify motor-based travel modes. Reported accuracy ranging from 74% to 100% depending on transportation mode.

[11] uses a three step classifier: 1) change point-based segmentation 2) inference model and 3) post-processing conditional algorithm considering the probability of transportation

mode transition between two adjacent segments. The authors, for the inference model, compared four different models where Decision Trees (DT) outperformed Bayesian Net (BN), SVM and Conditional Random Fields (CRF).

	Walk	**Car**	**Bus**	**Bicycle**
Walk	/	53,40%	32,80%	13,80%
Car	95,40%	/	2,80%	1,80%
Bus	95,20%	3,20%	/	1,60%
Bicycle	98,30%	1,70%	0%	/

Table 1 – Transition Probability matrix of transportation modes [11]

Table 1 from [11], reports that almost in all cases when travelling using Car, Bus and Bike transfer to Walking. Transition among other modes does not occur quite often.

[10] reported being able to distinguish between 10 different travel modes with an accuracy of around 91%. Their method is based on a fuzzy expert system to derive certainty factors for each transportation mode. Certainty factors are a system quantification of confidence based on evidence. Indicators used include speed variables, average proximities to bus and metro lines/stops and location of water courses to distinguish between land and water. GPS altitude was used to detect planes.

[14] approach detection of transportation mode evaluating metrics as travel duration, instant, mean, 95th percentile and standard deviation of speed and also the Rate of Change (RCM) in speed, where S_n is the nth speed measurement.

$$RCM = \sqrt{\sum_{n=1}^{N} \frac{(S_n - S_{n-1})^2}{N-1}}$$

[15] compare between five distinct classification models: 1) Naive Bayes (NB), 2) BN, 3) DT, 4) Random Forest (RF), 5) Multilayer Perceptron (ML). Also, the authors introduce in the list of features used to distinguish motor-based transportation modes: 1) closest Euclidean distance to rail line, 2) closest Euclidean distance to buses and 3) bus stop closeness rate. The authors introduce these features to solve the issue of motor-based transportation modes frequently having similar speed profiles. Also, the system has access to bus position in real time knowing this way if a determined GPS trace is coincident with the GPS position of the bus.

[12] uses a fuzzy engine to calculate the likelihood of each transportation mode using three variables: mean speed, 95th percentiles of speed and acceleration distributions.

2.6 Summary

In this chapter is reviewed the current state of art regarding clustering of spatiotemporal data, identification and meaning of Stay Point, the process to segment a trip and some relevant techniques to detect transportation modes.

To cluster information several algorithms are available. Algorithms like DBSCAN, DJ-Cluster and K-Means but simpler approaches to reckon stay points are often used.

The need to segment a GPS trace is acknowledged by researchers when research objective is related to detecting transportation mode; methods developed fall back on empirical knowledge like understanding need to stop or walk when changing between transportation modes.

	Modes	Segments	Classifier Model	Transition Probability	Length	Speed	Acceleration	Duration	Stop Time	RCM	Accuracy	GIS layers	Accuracy
[11]	4	Y	DT, BN, SVM, CRF	Y	Y	Y	Y	N	N	N	N	N	61,7%
[12]	5	Y	Fuzzy Engine	N	N	Y	Y	N	N	N	N	N	n.a.
[13]	5	Y	SVM	N	N	Y	Y	Y	Y	N	N	N	93%
[10]	10	Y	Fuzzy Engine	N	N	Y	Y	N	N	N	N	Y	91,6%
[14]	5	N	Cohen's Kappa Coefficient	No Seg.	N	Y	Y	Y	N	Y	N	N	80%
[15]	6	N	NB, BN, DT, RF, ML	No Seg.	N	Y	Y	N	N	Y	Y	Y	92,8%

Table 2 - Mode Detection Comparison

In order to achieve high accuracy detecting transportation mode, different approaches have been followed, with better or worse results depending on the features used, the methods applied to classify transportation mode and also depending on the studied population and respective GPS traces. The followed approaches commonly restrain to available data provided in a GPS trace that usually takes in account parameters like time, speed and distance. However, some authors reported using GIS layers to identify public transportation lines. The quality and quantity of GPS data available in each study also has significant impact in final results.

- [11] used a population of 45 individuals all with same GPS receiver model;
- [12] used a population of 4882 users during 6.65 days;
- [13] for example only tested their concept against 54 trips captured using handheld mobile devices in Hannover City;
- [10] used a dataset of 17 million points collected in the Netherlands and in other parts of Europe;
- [14] collected data of 12 volunteers with common work place;
- [15] data collection was extended only to 6 individuals over a 3 week period.

Techniques employed in researches have similarities between them in each of the presented subjects. Nevertheless, it is still missing a method capable of automatically classify transportation mode using clustered data instead of raw data. This issue that remains open will be addressed in Chapter 5.

Chapter 3

Problem, Technologies and Collection Tool

In this chapter it is provided a more complete overview of the problem explored in this thesis and it is made a review on the technologies applied. It is also important to describe the tool used to collect GPS traces, SenseMyFEUP and how the tool enable the collection of transportation mode used and GPS trace of a trip, both essential to train the developed algorithms.

3.1 Problem

In ubiquitous and context aware computing, understanding mobility of an individual from sensor data is an important area of research.

Analyzing urban mobility patterns allows retrieving information important to adapt existing infrastructures to the changing urban trends with expanding dimension and increasing complexity.

One of the most representative ways of describing mobility in an urban area is through origin-destination matrices. Origin-destination matrices are a fundamental source of information for traffic control and transport planning.

In order to make possible the automatic production of this matrices, it is the needed to improve data gathering and processing. One of the easiest and more complete ways of gathering this information is using smartphones' sensors that can retrieve contextual geographic information. It is important to consider that smartphones are everywhere nowadays, therefore, it is possible to take advantage of this information to make it useful to the community.

As retrieving origin and destination from GPS traces is a simple task, this thesis main goal aims to automatically classify transportation modes. Developing and training algorithms to detect transportation mode in a GPS trace requires evaluation of the trained algorithm. Training and evaluating the performance of a classifier is an essential but difficult task as it requires ground truth. In this case, it is needed to know that a GPS trace corresponds to a certain transportation mode.

3.2 Technologies

The implemented algorithms are written in PL/pgSQL and run in PostgreSql server. Doing the whole development in PL/pgSQL was an option made not only to reduce dependencies but also and essentially to increase performance, reducing network overhead and eliminating round trips. Moreover, it allows to quickly integrate the system with applications to use processed data without having the need to create APIs or to re-write complex algorithms in other programming languages.

Summarizing, main database is responsible for filtering, storing, processing and maintaining data. To complement PostgreSql features, extension PostGIS is used. PostGIS[1] adds support for geographic objects and provides essential features to work with spatial data. Features like built-in functions to interpret and process geographic information and the ability to create indexes based in geo-spatial characteristics.

3.3 Data Collection Tool

To collect data Instituto de Telecomunicações developed SenseMyFEUP, an Android application available at Google Play Store[2].

Figure 5 illustrates the data gathering architecture of the system used in this thesis.

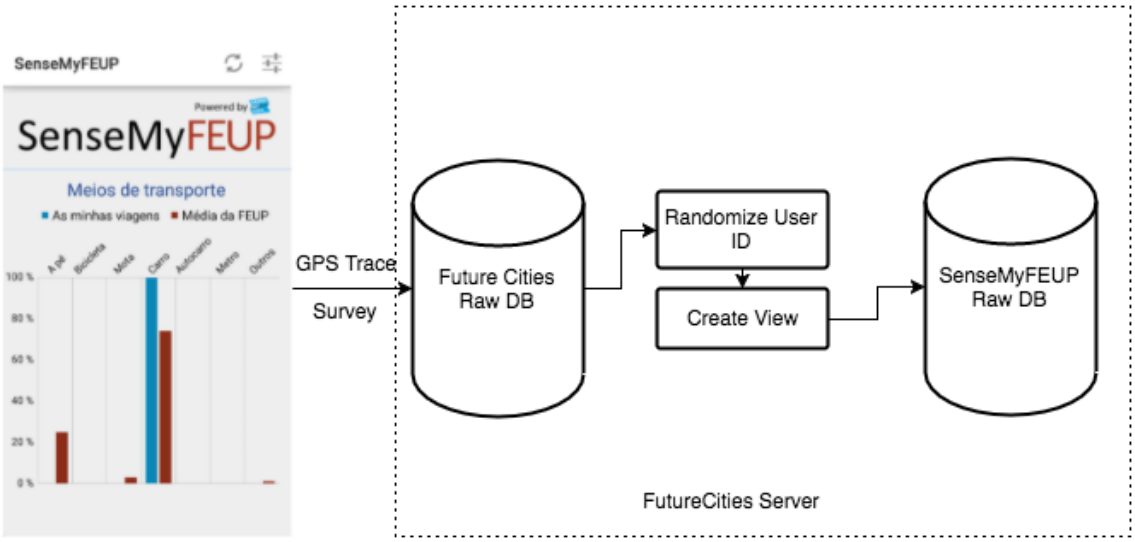


Figure 5 – Data Gathering Architecture

After SenseMyFEUP detects the end of trip, raw data is sent to server and it is filtered. This thesis starts in the moment raw is saved in database.

The application has an auto-start that senses movement to start collecting data. SenseMyFEUP detects automatically start and end of a trip based on the existence of significant movement within a certain interval of time.

Running in background often incomplete trips appear in system resulting by late recognizing of movement, Android memory management killing the background process or

[1] http://postgis.net/

[2] https://play.google.com/store/apps/details?id=future.cities.sensemyfeup

bad data resulting from Location Services in battery saving mode or with Location Services disabled. SenseMyFEUP does not obligate users to use High Accuracy on Android location detection settings, neither it forces to turn on Android Location Services[3].

The alternative to an auto-start feature would be the manual trigger of the collection process. Manual triggering is not optimal for development of ubiquitous applications, participants would often forget to start and finish the application. Therefore, automatically detecting existence of movement is a required feature to diminish user action leading to more data. However, this feature also has its weakness when the case is a pause exceeding stop thresholds, e.g. waiting for a bus that takes 10 minutes to arrive. Trip is flagged as complete. When user starts moving again, application will assume the start of a new trip.

In Figure 5 is possible to see a process called "Randomize user ID". The system is built addressing the arising concerns on keeping its users privacy safe, the whole system is built without knowing to whom each trip belongs, being the identification number randomized every 24h. To avoid changing the user identifier between two close (in time) trips from a participant, the routine to randomize the identifier is done at 4 am where few users will be travelling. This is important for trip chaining as it is shown in 5.4.

Besides raw data regarding GPS traces received from smartphones' sensor, it is also stored information received from surveys sent to user in the end of each trip. These surveys are essential to obtain a ground truth about transportation modes used during recorded session.

Figure 6 - SenseMyFEUP Survey

[3] https://developer.android.com/guide/topics/location/index.html

Figure 6 shows the survey sent to the user of SenseMyFEUP application everytime a trip is detected as finished. These surveys are used to train the models used to classify transportation mode. Surveys are the ground truth to automatically validate classified transportation modes and to measure the quality of each model in use.

It is possible to label Other ("Outros") as a mode, this is for not studied modes as boat, plane, train. Trips with this label will not be considered in the analysis and will be excluded from the dataset used for training the algorithms.

3.4 Summary

This chapter defined the problem that is proposed to be solved in this thesis regarding the construction of origin-destination matrices. Data collection and transportation mode detection in a GPS trace are the key subjects in analysis during the next chapters.

It is specified the technologies used to develop the algorithms that support the solution and it is presented SenseMyFEUP, the tool that supports data collection.

Chapter 4

Data Collection and Dataset

In this chapter the process of collecting data is described. It also gives an overview on the collected data and filters applied to narrow the information to the dataset used in the algorithms presented in Chapter 5 and to correct bad data.

The applied filters fit in two categories: 1) outlier detection and removal 2) filtering bad labelled data, both described in Section 4.2.

4.1 Data Collection

This thesis research is focused on the urban region of Porto and collected using smartphones equipped with GPS sensors, using Android operative system and a specific application developed for this thesis called SenseMyFEUP. The application and therefore the dataset were provided by Future Cities project from Instituto de Telecomunicações.

Data collection was registered in Comissão Nacional de Protecção de Dados (competent authority for matters related with data protection in Portugal). Process number 61.805.680.

To preserve participants' privacy, the raw data collected has a random identifier, assigned each day to each user. Therefore, in this thesis, it is only known if two trips were made by the same user if those trips occur in the same day, there is no possibility to directly identify multi-day patterns of mobility of a user.

Data collected can be classified into trip data (GPS dataset) and it was also recorded transportation mode using surveys in the end of each trip.

Students and teachers from FEUP are responsible for the majority of the collection however SenseMyFEUP is available in Google Play Store which make it virtually accessible to the whole world.

The period of collection are the months from April 2016 to June 2016 and involved 239 participants, produced 27 453 sessions (trips), recording 29 430 hours of commuting time and 289 624 km of travelled distance. In terms of surveys 7108 trips were labelled, 26% of the whole data set.

In all, it was recorded 2819 events of car, 106 of motorcycle, 3704 walking events, 732 events of bus commutes and 548 metro events, this values represent the answers from the

surveys and more than one event might belong to each trip since participants can choose more than one transportation mode per trip.

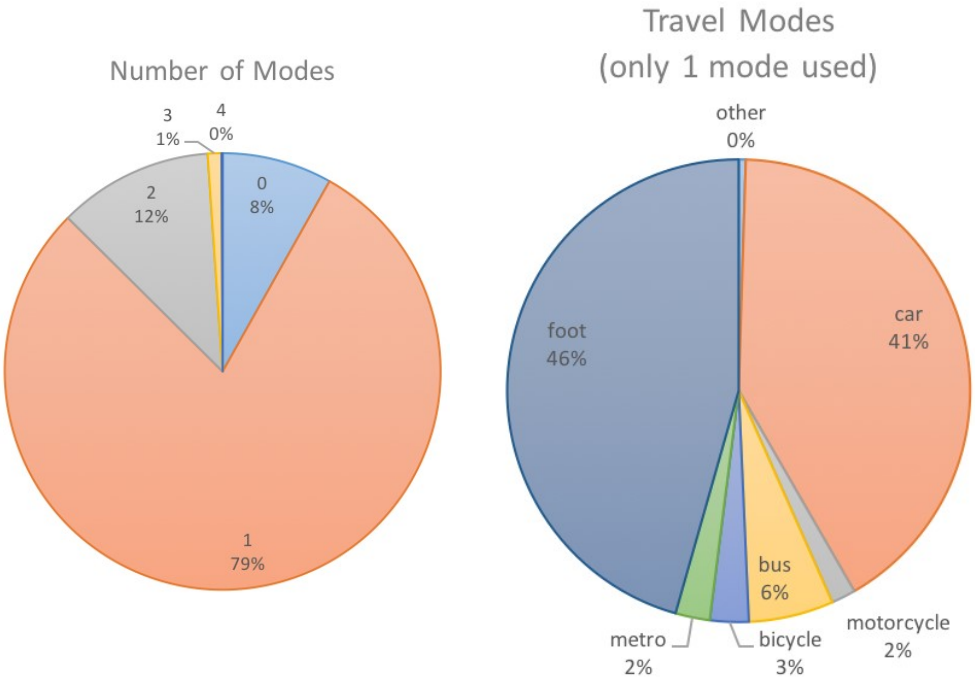


Figure 7 - Surveyed Travel Mode

In Figure 7, the leftmost chart pie shows how many modes were reported to be used in a single trip, usually 79% of the times only one mode was reported, 12% of surveyed trips report having more than one mode of transportation and only a few with more than two modes.

The rightmost chart pie represents the distribution of modes used when only one mode is reported, it is possible to verify that most of the trips were done by car or walking.

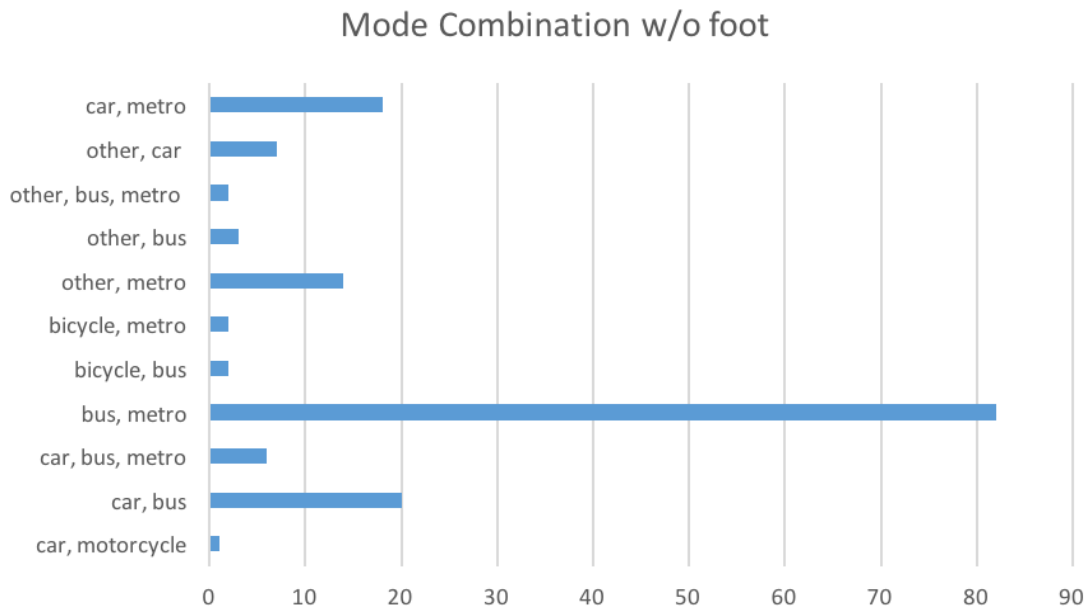


Figure 8 - Transportation Mode Combination

Observing Figure 8 it is possible to verify that, commonly, when a user reports using more than one transportation mode, both are in public transport category.

4.2 Data Cleaning

Using untrained participants as data collectors to record ground-truth data is not common in this type of researches, most authors use a research team to train systems. Using untrained participants poses a big challenge because trip logging is a monotonous, error-prone process by itself even more when working with prototype technology as it is SenseMyFEUP application.

Furthermore, this way research is not restricted to a small group of smartphones and quality of gathered data varies from smartphone to smartphone because different GPS sensors are used in each model.

Also collected data is exposed to different settings of location providers, having more or less precision and with more or less sample frequency.

However, despite these difficulties, with a population so broaden using and testing the application and algorithms, a strong perceiving of reality is provided as it is not just another lab simulation.

When working with data retrieved from sensors replicating real environments, several non-optimal observations may occur, sometimes more often than expected.

To achieve better results when processing data, some filters have to be used and algorithms have to be more resilient to errors. Before filtering data, it has to be assured that all needed parameters are present.

GPS sensor provides information relative to movement speed and altitude, however, speed is reported as 0 m/s and altitude as 0 m when location is not retrieved from GPS sensor but from cell tower triangulation, nearby Wi-Fi access points or any other method chose by Android Location Services different than GPS. When this condition occurs, point speed is calculated using the coordinates of the point in analysis and the previous point and their temporal difference and altitude is neglected.

The algorithm that generates stay points, as presented in Section 5.2, smooths singular errors. Clustering several correct points with few points with erroneous characteristics reduces the effect of erroneous point in system. As usually outliers have very high speeds, the algorithm works like a low pass filter.

Nevertheless, trusting error detection and filtering to Stay Points algorithm is not desirable as errors might be condensed in an area or occurring in burst leading to clusters of errors. To overcome this problem, a data filter is introduced in the system to clean raw data.

The data filter applied in this thesis focuses on two steps:

- Remove duplicate points. Two points are considered duplicated when having the same timestamp, latitude, longitude and speed but occurring on different milliseconds;
- Remove outliers. Outliers are evaluated regarding the distribution of speed; when working with geo-spatial data outliers can be identified by its very high instant speeds created by what can be described in visual terms as spatial jumps.

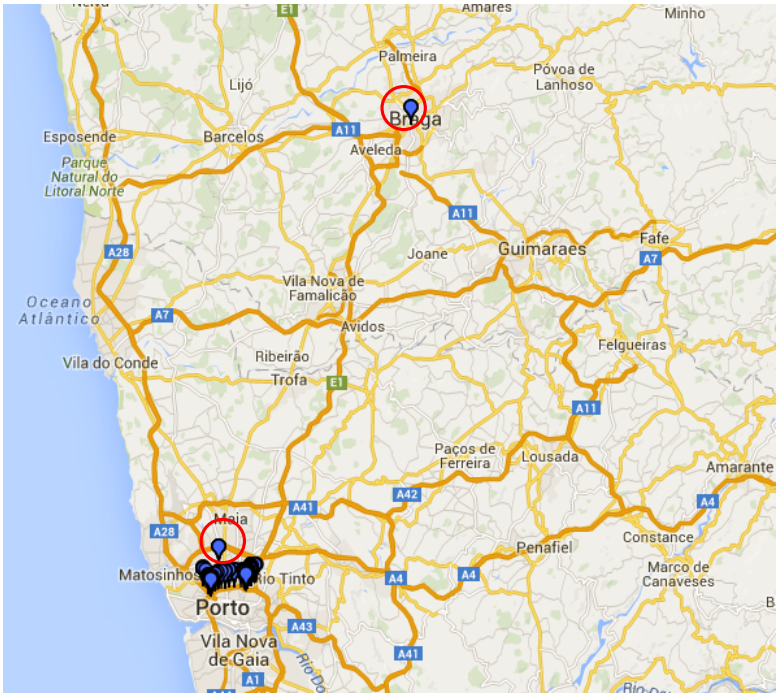


Figure 9 - Outlier Identification

In outlier detection some common heuristics are, for example, the three sigma rule ($\mu + 3\sigma$), the interquartile range analysis (Q3 + 1,5(Q3-Q1)) and the Median Absolute Deviation (MAD).

$$MAD = median(|Xi - median(X)|).$$

First heuristic (three sigma rule) is not used in this thesis because it does not adequate to speed type of distribution.

Three sigma method it is indicated for normal distributions (outliers included). Speed distribution in a trip is represented by a single sided heavy tail distribution, created by the extremely high speeds reported when a spatial jump occurs. Also, this indicator, that is supposed to guide the outlier detection, is altered itself by presence of outlying values because of standard deviation breakdown point being low. Breakdown point of an estimator is the ratio of incorrect values an estimator can handle before giving an inappropriate outcome.

Second method, Interquartile Range, it is a better approach when compared to the three sigma rule with a breakdown point of 25%.

Third method, MAD also called Hampel Detector, uses the median. Median is an indicator that measures central tendency. Moreover, Median is very insensitive to the presence of outliers. Median breakdown point is of 50%, MAD has the same behavior as median turning this heuristic the most interesting to use in this thesis.

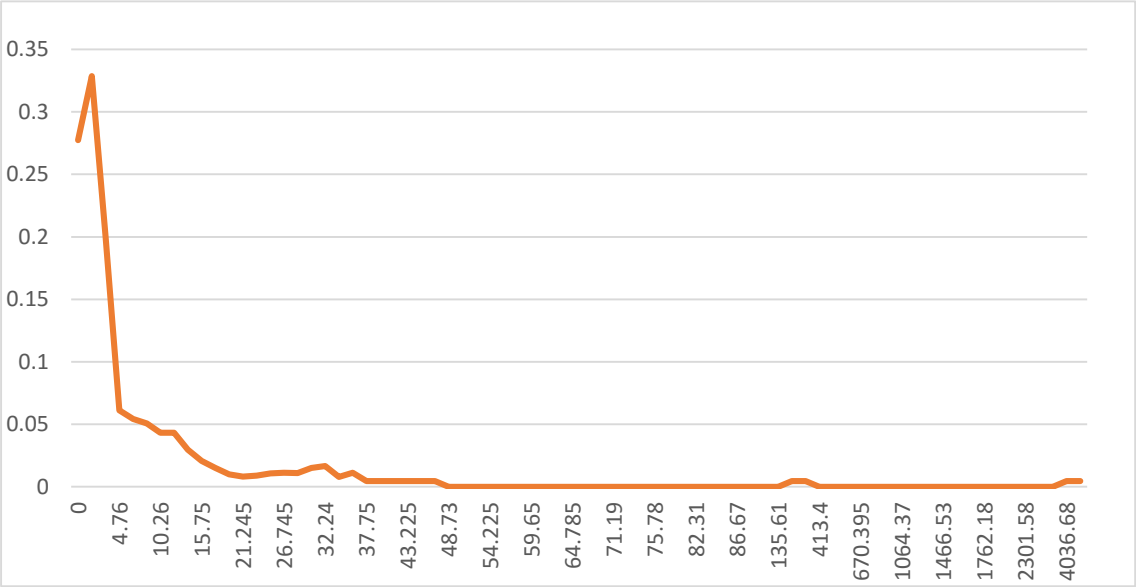


Figure 10 - Speed Distribution

Figure 10 shows a distribution of speed (m/s) and it is occurrence in an example trip. Analyzing this figure, it is possible to verify very high speeds that correspond to spatial jumps verified in trip, and can be seen in Figure 9.

When applying Hampel Detector to these distributions of speed outliers detected revealed to be very representative of the reality of possible errors and this way it is possible to eliminate most of the noise created by points resultant from bad data.

[16] suggest, as a generic definition, that a moderately conservative threshold should be around 2.5*MAD. Increasing the constant multiplied by MAD turns the formula more conservative, diminishing that value turns the formula more aggressive, detecting more possible outliers.

Studying the samples existent from collected data in speed distribution it was found that in order to only detect outliers a more conservative value should be used, around 6. Otherwise in a trip, e.g., starting in a highway and then entering in city center during rush hours, speed from the points recorded in highway could be identified as outliers.

Outliers in a trip are detected using the condition:

$$speed > \tilde{x} + 6 * MAD$$

After detecting the outliers, those points are marked and are not used in the determination of segments and stay points, consequently are not used in transportation mode detection.

While filtering outliers, anomalies are found and removed within each trip. Nonetheless, when working in a non-controlled environment as the one analyzed in this thesis, is important to keep an eye on data sanity.

A problem that often occurs, is bad labelling of data. Situations might be found of trips labelled as bicycle but tracked in a high-way at very high speeds, selection of metro where used transportation method was a train, or combined transportation and just one being

labelled resulting of users lack of understanding between the difference of a multi check box and a radio button (single option allowed).

Thus, in this thesis is done a sanity filtering of the data motivated by the question: “is this data physically possible?”.

For example, for some reason an answer to a survey might be just “Walking” and then in speed and route analysis is concluded that trip was done with a median speed of 70 km/h.

This has to be filtered and not considered when training methods or constructing statistical models.

For that reason, some filters have to be implemented to limit “impossible” situations. The filters applied are:

- Walk: Median speed < 10km/h AND Max speed < 20km/h
- Bicycle: Max speed < 50km/h
- Car: no limits imposed
- Metro: Max speed < 110km/h (Metro do Porto equipment speed limit + 10%)
- Bus: Max speed < 120km/h

With this filters in mind, bad labelled trips are, at least, limited to a few physically possible cases.

4.3 Dataset

As the area in analysis is Porto and as GIS layers available in system for metro and bus are only belonging to Porto, trips that do not start or end in a central area are discarded from the Training Data Set.

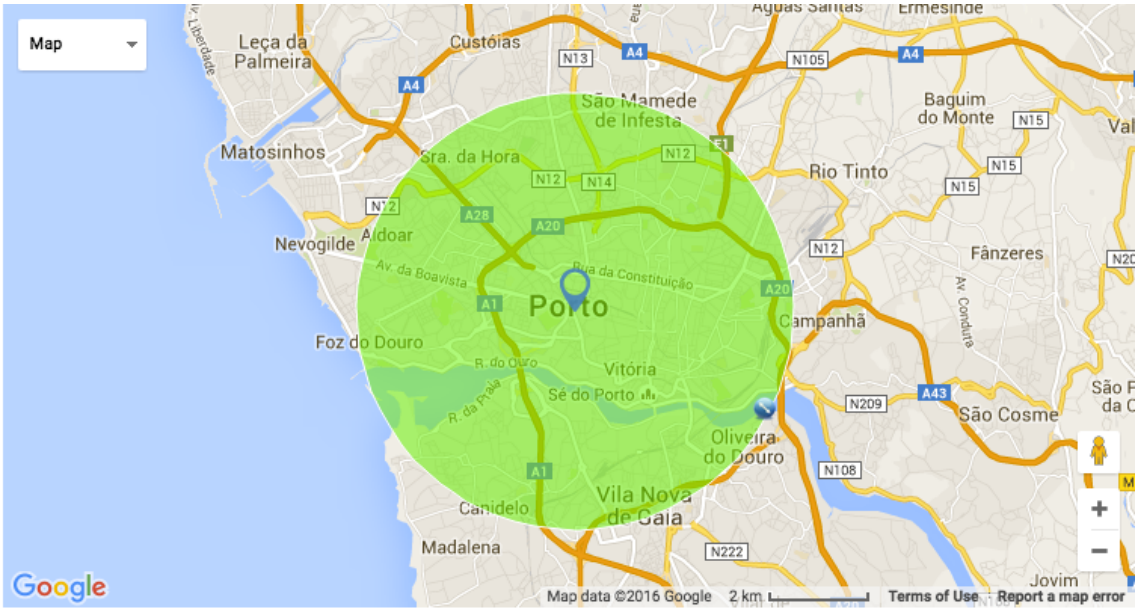


Figure 11 – Area restriction for Dataset

Figure 11 illustrates the area to which the dataset is restricted. Collected trips are filtered to obtain only the trips starting or finishing inside green circle. Green circle has a radius of 4km and center in city center.

Imposing spatial limits to the training set reduces analysis to 4008 surveyed trips. The dataset used for training the algorithms is done only on single transportation mode trips (ignoring walking as a trip when combined with another transportation mode). This limits reduce the number of trips to 2398. These are the trips used to train classifiers on speed features.

As motorcycle and car share a similar pattern of speed, both classes are joined under the class “Car”.

After filtering trips, everything is set up to train speed classifiers.

Boxplot comparison algorithm updates itself with each new filtered trip with a survey, additionally, only trips with just one reported transportation mode are used to train the algorithm since there is no information relative to which segments belong to each mode and it would create entropy in speed distributions determined.

Decision Tree training is done having in account not only speed distribution but also segment length.

In this thesis collected data, it is noticed different utilization of modes and, therefore, the amount of information regarding each transportation mode is different.

These big discrepancies on training set might inure the ability to generate a not over fitted decision tree. The training process might ignore classes (transportation modes) with less information as is the case of bicycle.

Class	Segments
bus	2688
foot	2913
metro	1554
bike	552
car	5100

Table 3 - Training Set

To avoid overfitting and to achieve reliable results, decision tree is trained with 10-fold cross validation. Cross validation allows the use of the whole subset in both learning and testing, dividing the example set into a number of folds.

In 10-fold cross validation the original dataset is divided randomly into 10 subsets of data, the folds. One fold is excluded when training the learning process and is used for testing the model. This process is repeated until all folds has been used at least once for testing. Then, the 10 results are averaged to produce a single estimation of the Decision Tree that is put to use.

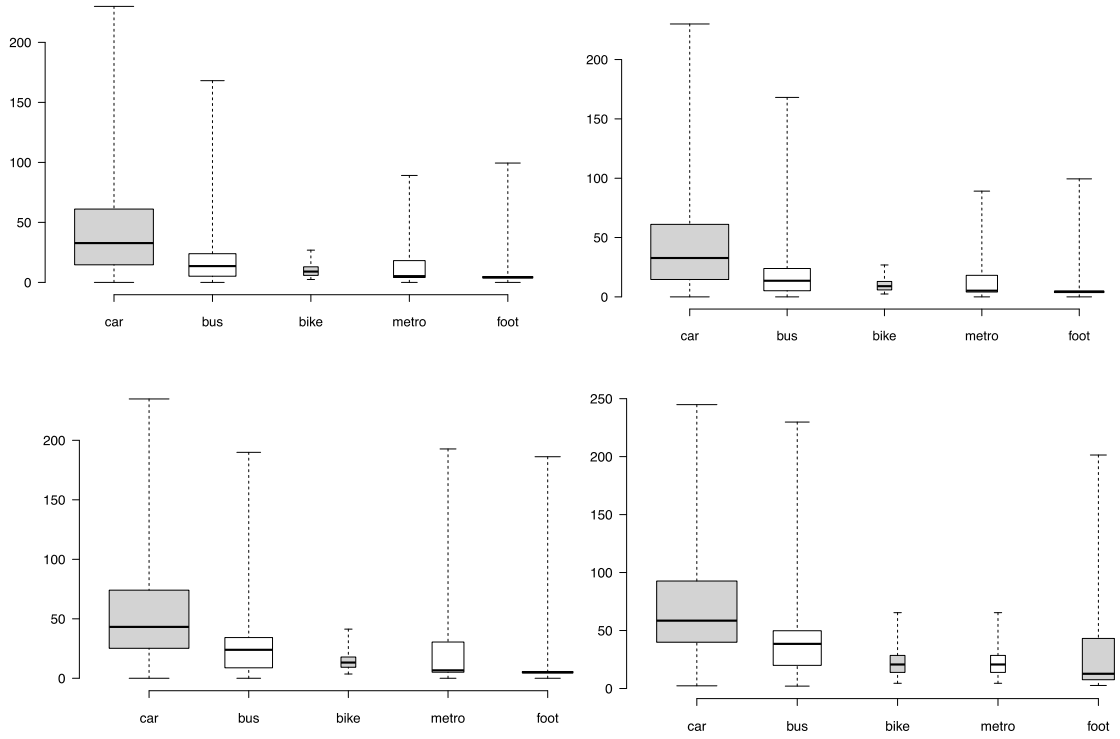


Figure 12 - Speed Distribution of Training Data

Figure 12 illustrates distribution of each component of speed saved to analysis for each transportation mode (y axis: speed is represented km/h). Upper left graphic represents first quartile distribution, upper right represents distribution of median, lower left shows distribution of third quartile and lower right illustrates distribution of max speeds.

It is important to refer that these graphics in Figure 12 are all created before applying sanity filters but with outliers removed, identified with conditions referred in Section 4.2.

4.4 Summary

This chapter describes the collected data and the process to clean trips from duplicated data and outliers. The process starts by filtering outliers using MAD heuristic and filtering trips identified as bad labelled, decision made by empirical defined speed thresholds.

The dataset is described and the restrictions made in order to restrain collected data, giving origin to the algorithms training data.

Chapter 5

Mode Detection on GPS Traces

This thesis focuses essentially in the development and integration of algorithms to analyze a dataset to find similarities in GPS traces to classify the transportation mode of each trip. These algorithms are detailed in this chapter by order of occurrence in the system.

When processing the information to automatically detect transportation mode from a GPS trace, the GPS trace is segmented and stay points are calculated originating a smaller representation of the initial trace. Each stay point is represented by information decided *a priori* as relevant to the transportation mode detection.

Trip transportation mode is classified using stay point information and classifiers are trained when a transportation mode survey is associated with the trip.

In the end of the process, a chaining of trips is performed, identifying trips from the same user that were detected by SenseMyFEUP as two separate trips. Trip chaining is necessary because this thesis problem is the offline treatment of crowdsourced GPS traces for extracting mobility information for transportation planning, concretely origin-destination matrices. These separate trips can be connected because, between them, time interval is within the defined thresholds and there is geographical coincidence of the end point and start point. This detected stops, that are chained, usually represent only a small stop in commute not interesting when studying mobility in an urban area.

5.1 Trip Segmentation

When analyzing the GPS trace of a regular commute trip, it is possible to face the situation of having, in a single trace, several transportation modes.

Thus, the trace needs to be segmented in sub-traces in order to allow detection of those multiple transportation modes possibly used.

Regarding trip segmentation, a basic assumption usually made and referred in the literature is called change point-based segmentation method. Change point-based segmentation: stop and walking is necessary when a change in the transportation mode occurs [13], [11], [12].

In the approach developed and applied there is no need to have a walking period to segment trip's GPS trace. It is sufficient to detect a stationary moment. These stationary moments will be candidate points to transportation mode change. Stops are used to segment trips. Stops are also candidate points to transportation mode change. Stop is detected according to the specification that to segment a GPS trace a speed threshold has to be lower than 0.5 m/s. Speed threshold is applied on the point being analyzed and the points belonging to the trace and within 5 seconds of the point being analyzed. The threshold measurement of speed is not applied to the absolute point speed but to the 80th percentile of speed in the analyzed point and having in account points in the following 5 seconds.

Segment "breaks" are useful not only to identify possible transportation mode changes but also to compare the stationary moments of a trip with bus and metro stops, an important metric to classify transportation modes used.

Input: GPS points with percentile80 of speed. timestamp: [t_{point}, t_{point}+5s] (ordered in time)
Output: GPS points with segment and movement information

```
WHILE point = next_point()
    IF percentile80_speed <= 0.5m/s THEN
            point_movement = false
            IF previous segment moving THEN
                    INSERT segment_point_list IN DB
                    new segment_point_list
            END IF
            append (segment_point_list, point)
    ELSE
            point_movement = true
            IF previous segment not moving THEN
                    INSERT segment_point_list IN DB
                    new segment_point_list
            END IF
            append (segment_point_list, point)
    END IF
ENDWHILE
```

Algorithm 1 – Trip Segmentation

After passing GPS raw data through trip segmentation algorithm data is marked with the information of movement and with the number of segment each point belongs to, each point can only belong to a segment.

Unfortunately, no solution can cover all situations that may occur in real life and this might fail when very fast transitions occur, transitions with less than 5 seconds stopped. Very fast transitions cannot be detected, e.g., a person running to a bus which instantly departs. Though, in practice this cases are not frequent. Changing speed and time thresholds to comprise this cases would affect not only the performance but also and largely the efficiency

of the algorithms. The benefits of possible modifications in thresholds would not bring more value to the proposed method.

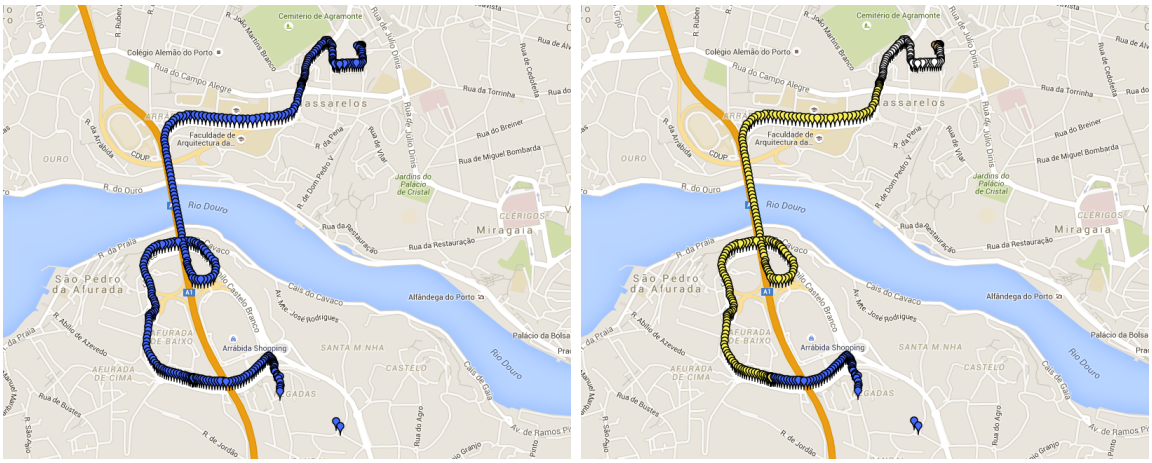


Figure 13 – Raw Trip and Segmented Trip

In Figure 13, raw data received from smartphone is illustrated on the leftmost map and on the rightmost map it is possible to see the different segments generated by the algorithm, each segment has a different color.

5.2 Reducing location trace information

After segmenting the trip to detect stationary moments and thinking on clustering data for future analysis, Stay Points have to be detected and characterized.

In Section 2.3, several solutions were presented on how to cluster geographic information, some based only on spatial density, some also considering time spent in a location.

The solution proposed in this thesis is based on the algorithm presented in [9]. Still, with some improvements in the way information is stored and the situations that are identified as a Stay Point.

When detecting stay points the main intention is to reduce the amount of data in analysis reducing several points description to a single-point with only relevant information representative of the whole clustered set.

To achieve this, first it is necessary to identify all the information that will be needed in further steps of process.

The set of features existent in literature used to classify transportation mode is vast. Studying the applicability of each feature combined with the use of stay points as source of data showed that clustered information might reduce the number of features available.

- **Acceleration**, when clustering GPS points the available sampling is reduced originating a cluster of points with the average information of those. This way the variation of speed between two clusters is not representative of the variation occurred during the trip. Internal analysis during clustering to maintain the information of acceleration distribution in cluster can be done. But it was proven to be a not good indicator with high and variable sampling rates as the ones provided by SenseMyFEUP application.

- **Stop rate,** stop rate is also lost with clustering. The maintained stop rate is regarding stops longer than 5 seconds which might be useful.
- **Travel time,** when studying an urban area and restraining the trips to city center, as described in 4.3, travel time is similar in the different motorized transportation modes not being useful to distinguish between them.
- **Head changing rate**, information of orientation is not provided by SenseMyFEUP.
- **Altitude,** altitude is not always reported as it depends if GPS sensor is available or if location is provided by Google Location Services.
- **Speed,** when clustering is possible to keep the distribution of speed of the clustered points maintaining this information available to posterior use.
- **GIS layers,** GIS layers can be put to use to identify closeness to bus and metro stops as they are detected by trip segmentation.

The second challenge is finding a method to represent this information in a single point maintaining it trustworthy and relevant.

In this thesis, Stay Points are defined as geographic regions, these are geographic regions that meet at least one of the following conditions:

1) defined as a physical region within a range D_{thres} between points where a user stayed for a certain amount of time T_{thres};
2) initial and last point of the trip;
3) points where a user is not in movement for a certain amount of time. These points are detected with trip segmentation algorithm and clustered with Stay Point algorithm.

Conditions 2) and 3) give origin to a stay point directly, those are two exceptions to the generic representation of a stay point identified by condition 1).

Generically, a set of points p_i = (lat, lon, T, speed) give origin to a Stay Point (sp) when within time (T_{thres}) and radius (D_{thres}) thresholds.

$$sp = \{p_m, p_{m+1}, \dots, p_n\} \; \forall \; m < i \leq n : haversine(p_m,\ p_i) \leq D_{thres} \wedge |p_n.T - \ p_m.T| \ \leq T_{thres}$$

A Stay Point is generated having its latitude, longitude coordinates in the geometric center (centroid) of the set of points *sp*.

$$\text{Centroid} = (C_{lat} = \frac{\sum_{i=m}^{n} p_i.lat}{|sp|}\ ,\ C_{lon} = \frac{\sum_{i=m}^{n} p_i.lon}{|sp|})$$

Besides the centroid, several other characteristics need to be saved to future use like average time ($T = \langle sp.T \rangle$), speed profile and distance between points that gave origin to the Stay Point. Speed profile is saved using the values needed to recreate efficiently a distribution of speed on that cluster.

This information is saved in the form of statistical descriptors of the speed with the values: min, max, mean, median, 25th and 75th percentiles of speed. The 25th percentile and 75th percentile, respectively are the lower and upper quartile of a distribution.

The statistical descriptor of speed used is called boxplot representation and is an efficient way of representing a distribution of each point (p_i) speed.

The box plot […] is a quick way to summarize the distribution of a dataset. In addition, this reduced representation […] provides a more straightforward way to compare datasets. - [17]

This will be a key factor in one of the algorithms presented in the next section to infer transportation mode.

Algorithm 2 shows the pseudo-code for the stay point algorithm developed in this thesis to cluster the GPS raw data into stay points.

```
Input: GPS points of segment with movement (ordered by time)
Output: stay points

first_point = ref_point = next_point()
APPEND (point_list, ref_point)
WHILE point = next_point()
    IF dist (point, ref_point) > Dthres
    AND interval (point, ref_point) > Tthres THEN
            CALL generateSP with point_list
            reset point_list
            ref_point = point
    ENDIF
    APPEND (point_list, point)
ENDWHILE

last_point = point
WHILE point = next_stop_point()
    IF dist (point, SP_moving) > 250m THEN
            CONTINUE
    ENDIF

    CALL generateSP with point
ENDWHILE

IF dist (first_point, firstSP) > 0.1 THEN
    CALL generateSP with first_point
ENDIF

IF dist (last_point, lastSP) > 0.1 THEN
    CALL generateSP with last_point
ENDIF

generateSP (point_list)
    COMPUTE centroid
    COMPUTE features
    INSERT in DB
end
```

Algorithm 2 – Stay Point

First the algorithm iterates through points belonging to segments previously categorized as moving segments, resulting in stay points. A similar behavior to the algorithm developed in [9]. However, the author does not consider the existence of segments in a trip. Not considering segments in a trip essential information to understand stationary moments is completely lost, information that might be useful to identify public transportation modes.

After representing trip movement in the form of stay points, the aggregated points representative of each stationary moment are clustered as a stay point.

Stay points from stationary moments have geographic center on the position of those points captured during the stationary moment. The assumed time of occurrence is the average time of occurrence of origin points. Stay points resulting from stationary moments need to be within 2*Dthres distance to at least one stay point clustered from moving points. This measure aims to protect from potential unfiltered noise. Data describes the approach to ignore outliers characterized by very high speeds. Although, outlier detection does nothing with low speed points that are related to ignored outlier points, when a “spatial jump” occurs very high speeds (outliers) are created. However, some other points might occur until signal reports again the correct position. During this time span points with zero speed might be created hypothetically, generating a stay point. Using the distance threshold to movement, it is prevented the creation of wrong stay points.

In the end of trip stay point calculation, it is verified if clusters near start and end points were created. If not, a cluster is created with the missing point (start or end) characteristics in order not to lose that information.

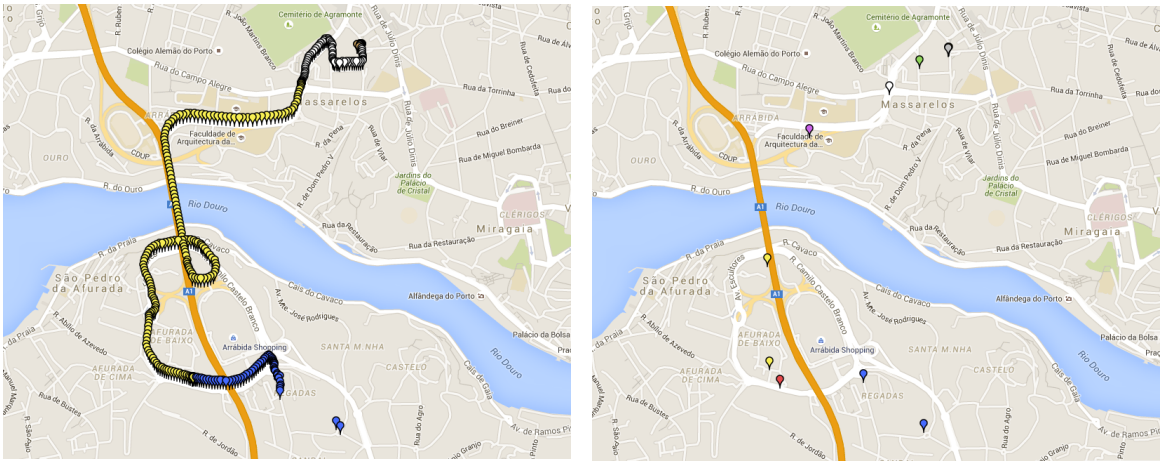


Figure 14 - Trip segments and stay points

Figure 14 represents two different stages of the process. In leftmost map representation, trip is segmented and different colors represent different detected segments. In rightmost map representation, points belonging to each segment are clustered in Stay Points using the algorithm described in Algorithm 2.

5.3 Detecting Transportation Mode

In this thesis is proposed a multi-stage classifier to overcome data uncertainties and also introducing a more robust process assessing transportation mode.

Classifiers based on simple rules such as speed and acceleration profiles cannot handle with great effect the analysis of a complete GPS trace due to reasons like:

- people often change their transportation mode during a trip;
- speed of different transportation methods is vulnerable to traffic conditions and signaling such as traffic lights and vertical signalization imposing low speed limits in an urban area.

To overcome the first problem mentioned above, literature review shows that the majority of authors prefer to begin the approach segmenting a trip to later discriminate transportation modes. Most use a walking segment to find a point where it should be possible to distinguish between two transportation modes. Using walking to identify possible mode change points is possible when using raw data. As in this implementation transportation mode detection is done on top of clustered data (stay points) it is not feasible to require walking segments between two transportation modes. The zero speed points that represent a stop will be clustered, and walking segments might be clustered in a moving segment belonging to any other transportation mode. Clustering points poses a limitation to the use of walking segments to detect possible transportation mode changes. Using walking segments would only be possible when in presence of a long enough walking segment that gives origin to, at least, one stay point.

Transportation mode detection is done with features present in stay points, stay points are grouped by segment. A segment has one or more stay points. Therefore, the smaller segment has to have at least one stay point.

While the core characteristic being evaluated to detect transportation mode is speed profile, other methods have to be exploited to resolve uncertainties.

Analyzing collected data shows that speed profile of cars and buses in the center of an urban area are very similar, then hard to distinguish. The second point enumerated before.

Also, movement patterns are hard to distinguish without extra information. Buses have to stop in bus stops although cars also have several stops either during rush hours or traffic lights. To overcome problems and ambiguities existent due to similarities in speed profiles and movement patterns of cars and buses, this thesis adds a GIS layer in the proposed multi-stage classifier.

Using a GIS layer, it is possible to incorporate geographic data corresponding to lines and stops of Sociedade de Transportes Colectivos do Porto (STCP) and Metro do Porto, the two main public transportation providers in targeted urban area of this thesis. Also, the only providers with information online from where it is possible to acquire precise line information.

Having limited information from public transportation lines may lead to bus trips not being identified as bus. Also, not being able to put a strong belief in bus network may lead to cars being identified as bus.

So contrary to metro analysis that produce a reliable final result, bus stop analysis produces just another variable to add value to final conclusions but that can be used by itself. Not only the lack of information makes the bus stop analysis not being able to be conclusive but also knowing that buses and cars may share the same route.

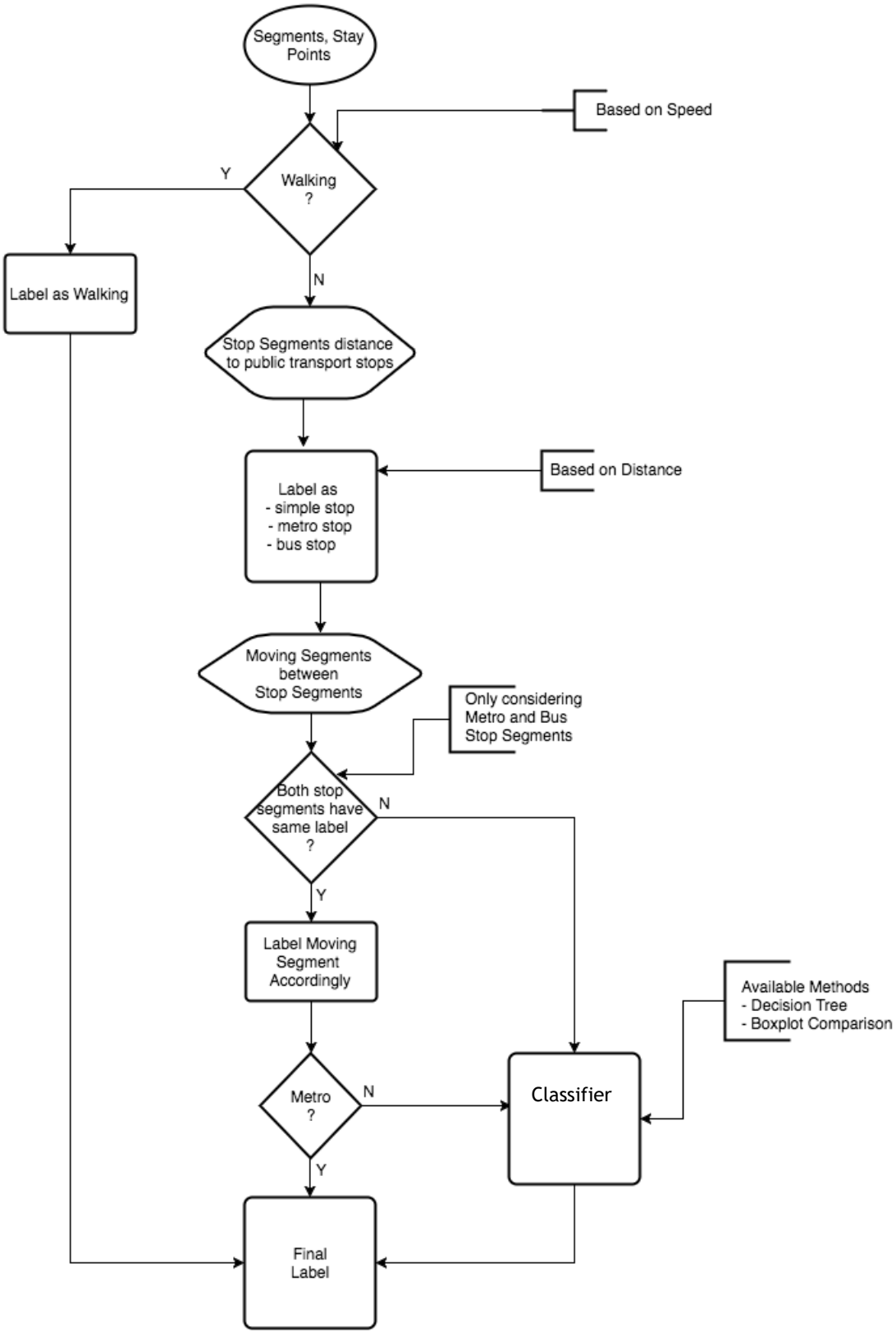
The developed classifier has a multi-step approach dividing problem in two main decision branches. One branch is responsible of detecting and classifying trip closeness to metro and bus lines, using stops information. The other branch takes care of trip speed profile, matching the trip being evaluated with previously learnt characteristics of the distribution of speed of each transportation mode.

Both modules converge its processed information in a decision engine that, using provided information, classifies the transportation mode used.

In this thesis to define the weight of each one of the two main pillars of information retrieved (speed profiles, closeness to bus stops and route, and closeness to metro stops and line) it is used a criteria based on entropy existing when applying each one to a test set.

Closeness to metro stops and line is the model that creates less entropy and where the data is more accurate. Lines are short and are not all around in the city like bus lines. Also, metro stops in every station, a bus may not stop. And cars can share pretty most all bus stops but few are the ones shared with metro.

Comparison of speed distributions is the model with more generated entropy due to sensibility to several external conditions like traffic, weather, manner of driving of each individual and road conditions.



Algorithm 3 - Transportation Mode Detection

Algorithm 3 starts with detecting all stay points that represent Walking.

It is assumed that a Walking stay point has 6km/h of median speed tops and that max speed is of 10km/h. These detected stay points are labelled as Walking.

Each segment without movement (represented by a single stay point) is compared to the GIS layer data to verify distance between stationary stay point and the closest bus station, and also distance to the closest metro station.

Thresholds are applied to identify not only the best candidate but also valid ones, e.g., there is no interest in assuming a metro stop far from the location, so all stops with a distance superior to a threshold are considered invalid possibilities. The defined threshold is 100m for both metro and bus stops.

When a stationary segment is not identified as bus or metro stop within 100m from the stay point it is labeled as “simpleStop”. Simple stops represent a stop that do not create value to decision.

Labeling stop segments first is crucial to this next stage. Moving segments (represented by one or more stay points) receive a label or not based on the characteristics of the limits of movement, i.e., if a moving segment is between two different metro/bus stops, segment is labelled as a candidate for being a metro/bus segment.

Particularly, in case of bus stops, both bus stops have to belong to same bus line. Otherwise, segments are considered just to have casually matched bus stops and are not labelled, by being not conclusive. In case of bus stops empirically is deduced that this might happen several times in an urban area and even might be a source of problems.

When a segment is not conclusive, segment speed profile is analyzed by a Decision Tree or a Boxplot Comparison algorithm. Both methods are used in this thesis in order to understand what better fits in the model, using one well reviewed and discussed in the literature (Decision Tree) and a new approach (Boxplot Comparison).

A decision engine (classifier) is used to classify the trip according to its transportation mode merging information processed by the two main branches: movement analysis and stop geo-data.

Both systems, Decision Tree and Boxplot Comparison, have a base of learning from existent data.

When learning speed profile of each transportation mode is important to understand that the same mode, in average, might have different profiles related to the length of the movement.

Using length to categorize a speed profile is critical as there is a tendency empirically understood to achieve higher velocities with longer distances traveled.

For example, knowing that a moving segment is created between two stop points, a moving segment created by a car in a small street with a length of, e.g., 500m hardly will achieve such high speeds as another segment of, e.g., 10km created in a highway.

Having this in account, learning system categorizes its learning process in transportation mode and length of segment.

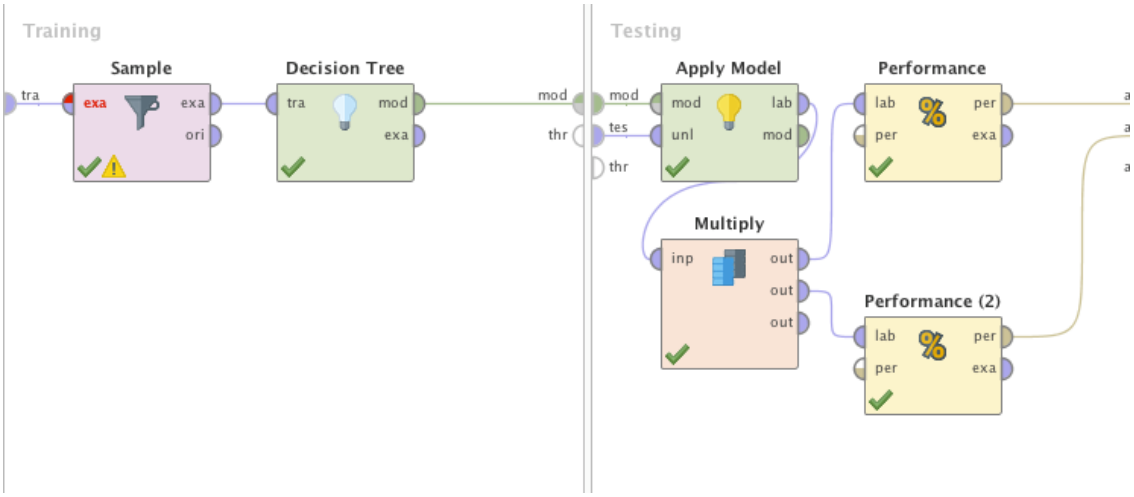


Figure 15 - Decision Tree

Figure 15 illustrates the scheme developed using RapidMiner Studio[4] to train the Decision Tree. Training set used is composed by the segments of each trip reported with only one transportation mode and labelled according to respective survey information.

Those segments are composed by stay points and have information of the speed distribution parameters calculated during stay point creation: first quartile, median, third quartile and max. It is also provided the length of the segment.

To equalize the quantity of segments of each mode used in training, a sub-sampling algorithm is put after retrieving the training data set. This sub-sampling picks the complete information and delivers subsets of information for training, up-sampling some classes (transportation mode) and down-sampling others achieving an average number of random samples provided to training algorithm.

Decision tree is trained with 10-fold cross validation. Cross validation allows the use of the whole subset in both learning and testing, dividing the example set into a number of folds.

When training decision tree it is not forced any use of the provided parameters, the system chooses what parameters to use to maximize gain ratios.

Boxplot comparison algorithm bases predictions on Euclidean distance between average boxplot parameters of each stay point of the segment being analyzed and average speed parameters learnt from ground-truth trips.

Training Boxplot method uses a different and simpler approach.

Boxplot parameters related to speed distribution are: first quartile, median, third quartile and maximum speed.

Each time a trip with survey is received, statistics are updated with new information, statistics are nothing more than the average of speed distributions grouped by transportation mode and segment length.

In boxplot training, the length intervals are well defined. Intervals have duration of 1km, starting in 0 until 10km, and after 10km all segments are merged in same category.

[4] https://rapidminer.com/products/studio/

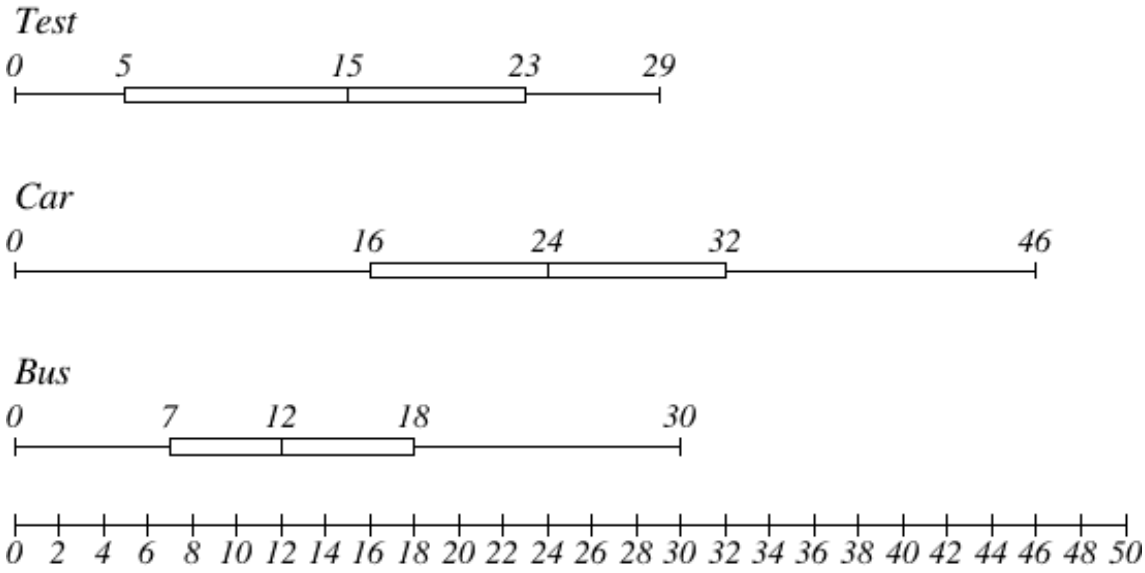


Figure 16 - Box Plot Comparison

Figure 16 illustrates boxplot comparison method, three boxplots are put "side by side" (in a graphical way of seeing it) and the in Test segment is analyzed against two ground-truth segments in the same length interval.

Knowing that Euclidean distance is given by:

$$d\,(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_n - p_n)^2}$$

Applying Euclidean distance definition example of Figure 16 to compare the three presented boxplots:

$$d\,(test, car) = \sqrt{\left(test_{\mathrm{Q1}} - car_{\mathrm{Q1}}\right)^2 + (test_{\mathrm{med}} - car_{\mathrm{med}})^2 + \left(test_{\mathrm{Q3}} - car_{\mathrm{Q3}}\right)^2 + (test_{Max} - car_{max})^2} = 23{,}92$$

$$d\,(test, bus) = \sqrt{\left(test_{\mathrm{Q1}} - bus_{\mathrm{Q1}}\right)^2 + (test_{\mathrm{med}} - bus_{\mathrm{med}})^2 + \left(test_{\mathrm{Q3}} - bus_{\mathrm{Q3}}\right)^2 + (test_{Max} - bus_{max})^2} = 6{,}24$$

As smaller Euclidean distance is from test and bus comparison, it is possible to conclude that test segment has higher chance of being a bus trip segment than a car trip segment.

After deducing conclusions related to stationary moments and to moving segments the system is ready to evaluate information available and classify trips' transportation mode.

Resuming to the analysis of Algorithm 3, from left branch are obtainable the walking segments, these segments maintain its label and do not require processing.

From central branch, is available stop information and data regarding movement between public transportation stops.

From right branch is delivered speed based information for moving segments not identified as segments related to public transportation.

Decision engine keeps information regarding stops without changes as it is not used to decide mode, just works as an auxiliary entry.

Decision engine works through iterations of acquired data. First iteration fills final results with walking segments, stop information and detected public transportation moving information. Second iteration goes through every segment, with missing data from previous iteration, and fills each with mode determined in speed analysis. Third iteration analyzes existence of specific situations like use of car and bicycle or car and bus in same trip. In these circumstances, it is determined speed profile for the whole trip and the one more consistent replaces previously determined values of both.

According to the existent dataset and to reviewed literature as exposed in Table 1 this is an abnormal situation and occurs more often in a not well observed speed estimative than in a real situation.

5.4 Trip Chaining

SenseMyFEUP has the feature of automatically detecting start and end of a trip based on the existence of significant movement within a certain interval of time. However, this feature also has its weakness when the case is a pause exceeding stop thresholds, e.g. waiting for a bus that takes 10 minutes to arrive. Trip is flagged as complete. When the participant starts moving again, the application will detect the start of a new trip.

For our goal, the origin-destination matrices, these trips should be a single commute. Thus, in this thesis was developed a method to detect these trips and join them, called trip chaining.

The algorithm restricts the search for same participant trips and to time and distance thresholds. For the chain to be detected two conditions must be verified:

- one trip must have start time within 30 minutes of previous trip end time;
- distance between the end and start of the two trips must be within a radius of 200 meters.

When two related trips are found a new entry relating both is created, this algorithm is a recursive algorithm and therefore finds all trips that meet these conditions.

A detected limitation to designed trip chaining algorithm is when SenseMyFEUP detects an end of trip due to a GPS signal shortage for a long enough period during travelling inside a tunnel, e.g. during a metro trip. In this cases chaining trips will not be possible because the distance criteria will not be met.

5.5 Summary

This chapter specifies the details of each algorithm used in the offline process to detect transportation mode on a GPS trace.

Trip is segmented having in account stationary moments, a stationary moment is defined as at least 5 seconds without movement.

Stay Points are clusters of points that belong to one of three conditions: 1) moving points not distancing more than 125m from each other and occurring in a 120s time interval 2) GPS

trace points belonging to each stationary segment give origin to a stay point 3) if there is no stay point generated comprising start or end points, those points generate a stay point.

Transportation mode detection is done on top of clustered data and two main characteristics of the trip: 1) closeness to public transport stops 2) speed distribution analysis – either using a boxplot comparison method or a decision tree. A decision engine is applied to infer the method when joining values calculated by in point 1) and 2).

A simple system to detect trips disconnected that would be joined in a spatial and temporal more broad way of looking at commutes is described in last sub-section of this chapter.

In this chapter it is possible to detect similarities and novelties facing the Literature. Some of the novelties presented in this chapter includes:

- Using segmentation to identify stops and use this information to create stay points;
- Detect transportation mode using stay point information, grouped by segment;
- Boxplot comparison to distinguish transportation mode speed profiles.

Chapter 6

Results

This chapter shows the performance of transportation mode classifiers implemented.

Additionally, it is important to evaluate quality of segmentation and stay point algorithms it terms of data reduction maintaining quality of data to be used by classifiers.

Classifiers are evaluated using three metrics, accuracy, precision (confidence) and recall (sensitivity):

1. Accuracy (TM) = (correctly classified instances of mode TM) / (total instances of mode TM) = (true positive + true negative) / (positive + negative instances)
2. Precision (TM) = (correctly predicted instances of mode TM) / (predicted instances of mode TM) = (true positive / true positive + false positive)
3. Recall (TM) = (correctly predicted instances of mode TM) / (real instances of mode TM) = (true positive / (true positive + false negative)

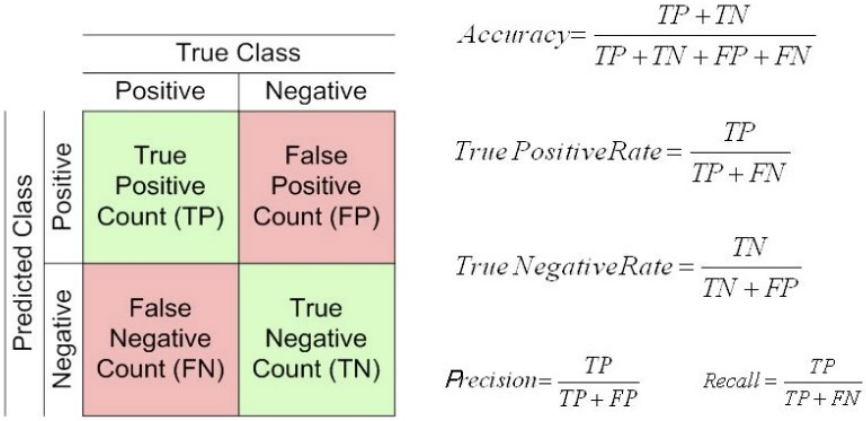


Figure 17 - Confusion Matrix Metrics [5]

Results are presented per mode and aggregated, analyzing the classes combined metrics.

Accuracy represents the percentage of predictions made correctly. A low precision number indicates many false positives. Recall represents the percentage of ground truth trips

[5] Source: http://images.slideplayer.com/24/7027794/slides/slide_60.jpg

correctly identified, thus a low recall number indicates that many ground truth trips were not correctly classified.

6.1 Data Cleaning and Compression

Analyzing collected data, it is verified that 19,47% of points are identified as duplicated and are discarded.

The number of outliers identified and removed is not easily calculated as the points are clustered and there is no information related to which points gave origin to each stay point. However, is possible to say the results are satisfactory due to lack of values with absurd values of speed caused by outliers.

Comparing initial number of points existent in raw data with number of stay points, from those trips, after clustering it is possible to verify a reduction to 1,65% (119066/7211291) of initial data.

6.2 Transportation Mode Classification

Results regarding classifiers are divided in two categories to enable comparison of the two classifiers: the results of Boxplot comparison algorithm and results from the analysis of the Decision Tree.

To understand how well classifiers are working, the question that is needed to answer is: “Was transportation mode A used in trip?”.

Results have to be validated against ground truth labels to verify if detected mode was used in trip.

When analyzing the performance of the classifiers of transportation mode, walking is not going to be considered. Walking is not considered as, virtually, all trips have walking segments and users do not mark walking as an used mode in a trip if the walking part of the trip is reduced, e.g., walking from the parking lot to faculty.

	Accuracy	Precision	Recall
Metro	96,38%	80,82%	91,24%
Car	91,60%	93,11%	94,13%
Bus	89,13%	67,64%	84,79%
Bike	93,30%	22,66%	55,77%

Table 4 – Metrics for Single Mode Trips, Boxplot comparison

Algorithm 3 when combined with boxplot algorithm performs very well for driving (recall 94,13%) while not raising many false positives (precision 93,11%).

Metro activities have a high percentage of detection (recall 91,24%) but raising some false positives (precision 80,82%).

Despite the good performance regarding bus transportation (recall 84,79%), the algorithm raised many false positives (precision 67,64%). Within a bus activity, very often trips occur in city center where bus have to move at slow speeds behaving like bicycles. Other times is verified that buses that travel from or to outside of Porto present high speeds as a car, causing frequent confusion between these classes.

Bicycle presents the worse metric results from the four classes not performing well (recall 55,77%) and raising too many false positives (precision 22,66%).

These unsatisfactory results were originated by the reduced number of existing bicycle trips. The number of bicycle trips is small compared to other transportation modes. This might be related to the configuration of the urban area in analysis, an area of hills and few streets without elevation gains and elevation losses.

Results show that with this classifier it is possible to distinguish between different mobility states with high accuracy, having a problem detecting bicycle. The algorithm presents an overall accuracy of 92,60%, an overall precision of 81,56% and an overall recall of 90,86%.

	Accuracy	Precision	Recall
Metro	91,26%	58,62%	85,78%
Car	70,76%	71,27%	91,13%
Bus	67,01%	25,84%	39,15%
Bike	93,78%	9,09%	3,37%

Table 5 - Metrics for Single Mode Trips, Decision Tree

Classification schema when using Decision Tree algorithm have more disparate results than with Boxplot algorithm.

The classifier performs well for driving (recall 91,13%) with some false positives being detected (precision 71,27%). Metro has a good performance (recall 85,78%) raising several false positives (precision 58,62%). Bus has a poor performance (recall 39,15%) and with numerous false positives being detected (precision 25,84%). Also in this combination of algorithms bicycle is the class that perform worse (recall 3,37%) and too many false positives being detected (precision 9,09%).

High values of accuracy with lower precision and recall values shows that classifier is assuming well the values that do not belong to the class, however, does not detect correctly trip mode. The method presents an overall accuracy of 80,70%, an overall recall of 76,36% and an overall precision of 58,64%.

6.3 Discussion

Results show that when using Decision Tree in detection schema quality of inference is diminished compared to using Boxplot method.

Comparing results obtained in this thesis, when using boxplot comparison within Algorithm 3, with results obtained in several researches existent in Literature it is possible to assume that this thesis presents a method to classify transportation mode from GPS traces. The presented methods have a similar performance to that of other methods found in the literature.

- [11] compared several methods on top of change point based segmentation but non-clustered GPS trajectories. The authors report best method as being analyzing accuracy by length with a uniform length of 150m (on segments). The overall accuracy of mode detection (walking, car, bus, bike) stated is 61,7%;

- [13] reports 93% of classification accuracy;
- [10] reports an overall accuracy of classification of 91,6% to classify between 10 different modes. The authors use GIS layers with road information, bus/metro/tram/train lines and also they made use of altitude to find planes;
- [14] report 80% of overall accuracy in detecting between 5 transportation modes (walk, bike, train, bus, car);
- [15] report an overall accuracy of 92,8% and an overall recall of 92,9% when considering transportation network data (public transportation lines and real time location of buses).

It is also important to notice that several Literature study populations were not as complex as the studied population in this thesis.

- [11] used a population of 45 individuals all with same GPS receiver model;
- [13] for example only tested their concept against 54 trips;
- [10] used a dataset of 17 million points collected in the Netherlands;
- [14] collected data of 12 volunteers;
- [15] data collection was extended only to 6 individuals.

Looking at results obtained from other researches and identifying population in study in each one and methods applied is critical to understand how well the proposed algorithm is performing.

Evaluating differences between literature results and methods, and the ones followed in this dissertation is important to learn what to improve in future work in order to improve existent results.

Boxplot comparison method has proved to be a good solution with low computational cost rivalling with the best results obtained in reviewed researches, even when using a more complex collection of data.

Also, it is important to recognize that the algorithm proposed analysis clustered data and reviewed researches work on top of raw data and use additional data sources which provides more information.

Chapter 7

Conclusion

In this dissertation, we started by surveying the existent literature to understand the state of the art.

A multi-stage solution to extract mobility relevant information from crowdsourced data to create origin-destination matrices is proposed.

Data filtering, successfully found and ignored errors in surveys and outlier points resulting from problems with location provider making data evaluated more smooth and clean.

Segmenting GPS traces based on stationary moments in a trip proved to be an effective way to find important contextual information. This information allows to create a relationship between trip stationary moments and bus and metro stops.

Stay points, generated from either moving segments and stationary ones, revealed being an effective way of clustering GPS points reducing raw GPS traces to 1,65% of original data preserving essential information to be used in tasks like detecting transportation mode.

An essential indicator regarding mobility is transportation mode used in commute, therefore, the process culminates in a system to automatic classify transportation mode used in a recorded trip consisting of a GPS trace.

To classify transportation mode, an algorithm was developed having its core auxiliary to decision divided into two main branches: 1) stationary moments close to bus and metro stops (closeness to public transportation) 2) speed distribution analysis. Combining results from these two branches into a decision engine.

To classify transportation mode based on speed features, two approaches were followed to analyze speed distribution profiles.

The approaches followed were a Decision Tree and an approach not-known to be used to solve this type of problems. The approach consists in a system to compare boxplots of clustered point speed distributions.

The boxplot comparison algorithm evidenced results of good performance when compared to Decision Tree. Moreover, when comparing results obtained using the boxplot comparison algorithm to results achieved by reviewed researches in Literature. Another advantage of the developed algorithm is being a continuously learning algorithm representing low computational cost compared to methods frequently used to detect transportation mode.

7.1 Contributions

The following items summarize the main contributions of this thesis:

- **Support data collection campaign**. To foster engagement of participants in the data collection process information has to be provided. Metrics are sent back to the users regarding their statistics and global statistics of the participants.
- **Development of procedures to clean the dataset**. One of the hardest and more time consuming tasks faced when working with real-world data is identifying and cleaning bad data. Methods were developed to identify and clean wrong labelled trips and bad data present in trips resultant from location sensor misbehavior.
- **Stay Points as data source for transportation mode classifiers**. This thesis introduces the capacity to join several, for long studied, concepts never used in a joint solution. Themes as Trip Segmentation, clustering points into Stay Points and Transportation Mode Detection have been exhaustively studied independently. Often trip segmentation and transportation mode detection are used together. However, for the first time stay points are generated having in account segments and are used as data source for transportation mode detection.
- **Boxplot comparison algorithm to classify transportation mode.** New method to understand speed relation between a trip and the trained speed profiles of each mode is proposed. The method exhibits good performance, low computational cost and ability to continuously adapt to the system.
- **Collaboration with the Urban Planning team.** Supported the collaboration with the Urban Planning team providing datasets used to analyze the mobility patterns of the studied population and validate them against the actual methods of studying mobility.

7.2 Future Work

In order to better understand collected data and the obtained results a statistical analysis of this information would provide a better overview of the algorithms performance and of the collected sample. However, this would need to, in some way, reduce the privacy of the users during the analysis.

To improve Stay Point algorithm, duration of each stationary stay point could be added to saved information.

Future work in area of transportation mode detection, includes exploring more knowledge regarding metrics that might be put to use to complement transportation mode classifier. One metric that might be a good candidate to complement the system is the number of stay points generated in each trip (by distance). Different transportation modes present different quantity of stops per kilometer, e.g. in a same street a bus will stop more times than a car because of bus stops, therefore the number of segments generated will be varying. A car in a highway will generate more moving stay points than stationary ones.

	Stop	Moving	Total
bike	3,54	2,88	6,42
bus	5,23	2,61	7,84
car	2,04	1,41	3,45
foot	0,91	0,92	1,83
metro	6,34	2,51	8,86
other	3,97	1,83	5,81

Table 6 - Average Stay Points per Km

Analyzing the statistics retrieved from gathered data and displayed in Table 6 it is possible to verify that public transports have a bigger amount of stops per kilometer than bike, car and walking. With this heuristic in account it is possible to weight decision when in doubt between, e.g., car and bus or use it as a feature of the classifier.

As presented in Literature, a transition probability matrix of transportation mode should be added. To do so distances between boxplots and distance to public transportation stops now being evaluated should be converted to probabilities to then calculate final probability of a segment correspond to each mode.

This metric cannot be used in presented implementation without big changes in behavior, because this thesis approach to the problem works by segment, and this would be a distinguisher that could be used for the whole trip only. The present system is only ready to receive metrics (as modules) if they are metrics by segment.

An algorithm to fix problems with auto-stop feature from SenseMyFEUP was developed (Section 5.4) but results were not used to train and test the classifier. An improvement would be to classify trips after the trip chaining algorithm and not before, having in account the existing labels of one or more trips merged. Moreover, during tests and development of trip chaining algorithm it was detected that SenseMyFEUP in some smartphones stopped recording a trip in a metro tunnel, giving origin to two trips. One before entering the tunnel and another after the tunnel. This situation is not predicted by trip chaining algorithm, a modification to include this cases could be designed being restrained to trips close to metro lines. Metro lines are available in GIS layers.

Scaling the developed system in this dissertation until a certain number of users is not problematic. All algorithms have an efficient performance because of its reduced complexity. However, in order to make it scalable some tasks have to be, even more, independent from each other. For example, updating boxplot statistics should be done from an outside call and not in the sequence of processing trip.

Bibliography

1. Ester, M., et al. *A density-based algorithm for discovering clusters in large spatial databases with noise*. in *Kdd*. 1996.

2. Birant, D. and A. Kut, *ST-DBSCAN: An algorithm for clustering spatial–temporal data.* Data & Knowledge Engineering, 2007. **60**(1): p. 208-221.

3. Zhou, C., et al. *Mining personally important places from GPS tracks*. in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. 2007. IEEE.

4. Zhou, C., et al. *Discovering personal gazetteers: an interactive clustering approach*. in *Proceedings of the 12th annual ACM international workshop on Geographic information systems*. 2004. ACM.

5. Zignani, M. and S. Gaito, *Extracting human mobility patterns from gps-based traces.* 2010: IEEE. 1-5 %@ 1424492300.

6. Ashbrook, D. and T. Starner, *Using GPS to learn significant locations and predict movement across multiple users.* Personal and Ubiquitous Computing, 2003. **7**(5): p. 275-286.

7. Zheng, Y., et al., *Mining interesting locations and travel sequences from GPS trajectories*. 2009: ACM. 791-800 %@ 1605584878.

8. Li, Q., et al. *Mining user similarity based on location history*. in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. 2008. ACM.

9. Silva, T.A.C.d., *Data mining de dados geo-temporais para suporte à mobilidade*. 2013.

10. Biljecki, F., H. Ledoux, and P. Van Oosterom, *Transportation mode-based segmentation and classification of movement trajectories.* International Journal of Geographical Information Science, 2013. **27**(2): p. 385-407.

11. Zheng, Y., et al. *Learning transportation mode from raw gps data for geographic applications on the web*. in *Proceedings of the 17th international conference on World Wide Web*. 2008. ACM.

12. Schuessler, N. and K. Axhausen, *Processing raw data from global positioning systems without additional information.* Transportation Research Record: Journal of the Transportation Research Board, 2009(2105): p. 28-36.

13. Zhang, L., et al. *Multi-stage approach to travel-mode segmentation and classification of GPS traces*. in *Proceedings of the ISPRS Guilin 2011 Workshop on International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Guilin, China*. 2011.

14. Huss, A., et al., *Using GPS-derived speed patterns for recognition of transport modes in adults.* International journal of health geographics, 2014. **13**(1): p. 1.

15. Stenneth, L., et al. *Transportation mode detection using mobile phones and GIS information*. in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2011. ACM.

16. Leys, C., et al., *Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median.* Journal of Experimental Social Psychology, 2013. **49**(4): p. 764-766.

17. Potter, K., et al., *Methods for presenting statistical information: The box plot.* Visualization of Large and Unstructured Data Sets, s, 2006. **4**: p. 97-106.