WISSENSCHAFTLICHE ARBEITEN DER FACHRICHTUNG
GEODÄSIE UND GEOINFORMATIK DER LEIBNIZ UNIVERSITÄT HANNOVER

Dissertation

Mining GPS-Trajectory Data for Map Refinement and Behavior Detection

Lijuan Zhang

HANNOVER 2014

Erklärung zum Promotionsgesuch

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig verfasst habe, die benutzten Hilfsmittel vollständig angegeben habe und die Dissertation nicht als Diplomarbeit, Masterarbeit oder andere Prüfungsarbeit verwendet habe. Weiterhin erkläre ich, dass ich keine anderen Promotionsgesuche eingereicht habe.

Summary

In today’s world, we have increasingly sophisticated means to record the movement of moving objects such as vehicles, humans and animals in the form of spatio-temporal trajectory data. As a consequence of this development, increasing volumes of such data are being accumulated at an extremely fast rate. A trajectory is usually represented by an array of structured positions in space and time, i.e. each has a signature of specific location (geospatial coordinate tags) in time (time stamp tags). The data hold information about the representation of spatial phenomena, such as the geometry of the environment. Moreover, they also provide information about the spatio-temporal behavior of the moving objects. As a result, knowledge discovery from these data has become an important problem and is in increasing demand to understand the underlying nature of the data, and it aids in various decision-making processes.

In the thesis, different approaches for the analysis of trajectories in the context of navigation and location based services are presented: the determination of the geometry of a road network, the classification of the travel mode of the moving objects, as well as the identification of different types of behavior such as anomalous driving patterns.

A novel method towards improvement of existing OSM road data from incoming, massive amounts of GPS-trajectory data is presented. We use the OSM road map as a reference map and match GPS-trajectories with corresponding roads using both geometrical and statistical method. Matching according to their travel modes is also applied to tackle errors in GPS data. We also mine additional attribute information from such data.

It is also possible to infer the travel mode from the trajectories. This can be used to identify the road type from which the GPS-trajectory is collected, and thus allows to also add semantic attributes to the geometries of the roads extracted. Other location-based services could also benefit from such information. We take six travel modes into consideration, which supposedly consists of different movement-patterns: walk, bicycle, car, bus, tram and train. A two stage classification method is developed to robustly detect travel mode from trajectories. Due to the fact that GPS trajectories are often composed of more than one travel modes, they are firstly segmented as movement segments by identifying stops, which are classified as pedestrian, bicycle, and motorized vehicles to find sub-trajectories that corresponding to individual travel modes. In the second stage, a breakdown classification of the motorized vehicles class as car, bus, tram and train is implemented based on sub-trajectories using Support Vector Machines (SVMs) method.

Trajectories also reflect the behavior of the subjects producing them. In the thesis, several examples are given about possible behavior patterns. One is the anomalous driving behaviors detection. It is of high interest for applications in the areas of navigation/driver assistance systems, surveillance and encountering navigation problems, i.e., taking a wrong road, performing a detour or tending to lose the way. An extended Markov chain is used to remodel the trajectory. And a recursive Bayesian estimator is conducted to process the Markov model and deliver an optimal probability distribution of the potential anomalous drive behaviors dynamically over time.

Zusammenfassung

Heutzutage stehen immer mehr Möglichkeiten zur Verfügung, um bewegte Objekte wie Fahrzeuge, Menschen und Tiere in Form von räumlich-zeitlichen Trajektorien zu erfassen bzw. zu dokumentieren . Als Folge dieser Technologieentwicklung können solche Trajektorien-Daten mit hoher räumlicher und zeitlicher Auflösung gesammelt werden. Eine Trajektorie wird normalerweis in Form eines Vektors mit Koordinate und Zeitstempel dargestellt. Die Daten enthalten explizite und implizite Informationen über räumliche Phänomene, wie z.B. die Geometrie der Umgebung. Darüber hinaus enthalten sie Informationen über das räumliche und zeitliche Verhalten der bewegten Objekte. Infolgedessen ist die Analyse dieser Geodaten zu einer wichtigen Forschungsfrage geworden; hiermit lassen sich nicht nur wichtige Informationen über die Objekte und ihre Umgebung gewinnen, sondern auch als Grundlage von Entscheidungen nutzen.

In der Arbeit werden verschiedene Ansätze der Trajektorienanalyse im Kontext von Navigation und Ortsbezogenen Diensten (LBS) vorgestellt. Es sind dies insbesondere ein Ansatz zur Bestimmung der Geometrie des Straßennetzes, die Klassifikation des Fortbewegungsmodus‘, sowie die Detektion von Verhalten, insbesondere anormalem Verhalten im Straßenverkehr.

Eine neue Methode zur schrittweisen Verbesserung von bestehenden Open Street Map (OSM) Daten anhand von GPS-Trajektorien wird vorgestellt. Die Straßen in OSM werden als Referenz der GPS-Trajektorien zu korrespondierenden Straßen betrachtet, um Unsicherheiten und fehlerhafte Daten in den GPS-Aufzeichnung durch geometrisch statistische Verfahren zu ermitteln. Zusätzlich werden weitere Information wie die Anzahl der Fahrspuren aus den GPS-Trajektorien erfasst.

Aus den Trajektorien können weiterhin Informationen über die Fortbewegungsart der Objekte ermittelt werden. Dies kann z.B. genutzt werden, um eine korrekte Identifikation des Straßentyps aus GPS-Trajektorien, und damit eine semantische Information, zu gewinnen. Dies hat z.B. große Bedeutung für ortsbezogene Dienste (LBS). In der Arbeit werden sechs Fortbewegungsarten in Fußgänger, Fahrrad, Auto, Bus, Straßenbahn, Zug unterteilt. In diesem Fall können die GPS-Spuren in Abhängigkeit der Fortbewegungsart in verschiedene Sub-Trajektorien segmentiert werden. In der ersten Klassifizierung der Sub-Trajektorien nach ihrer Fortbewegungsart handelt sich um Fußgänger, Fahrrad und Auto. Dabei werden die GPS-Spuren hinsichtlich Geschwindigkeit, Beschleunigung und Richtung der Bewegung segmentiert. Die zweiten Klassifizierung der Sub-Trajektorien erfolgt anhand Support Vector Machines (SVM). Dabei bezieht sich auf Auto, Bus, Straßenbahn und Zug im Vergleich zur ersten Klassifizierung.

Trajektorien spiegeln das Verhalten der bewegten Objekte wider. In der Arbeit werden einige Anwendungen im Kontext der Verhaltenserkennung dargelegt. Insbesondere wird ein Ansatz zur Erkennung von anomalem Fahrverhalten vorgestellt, der von großer Bedeutung ist für Anwendungen in den Bereichen Navigation bzw. Fahrerassistenzsysteme, Überwachung und Notfallmanagement. In dieser Arbeit erfolgt eine Konzentration auf GPS-Trajektorien und Identifikation der Zeitpunkte und Orte, wo der Fahrer Navigationsproblemen hat, d.h. sich verfahren hat oder einen Umweg fährt. Eine erweiterte Markov-Kette wird verwendet, um die Trajektorie mit der Integration langfristiger Eigenschaften umzumodellieren. Anschließend wird eine rekursive Bayes-Schätzung durchgeführt, um das Markov-Modell zu verarbeiten, und eine optimale Wahrscheinlichkeitsverteilung des möglichen anomalen Verhaltens dynamisch über die Zeit zu liefern.

Contents

1 Introduction 1
1.1 Motivation 1
1.2 Road map refinement from GPS-trajectory data 2
1.3 Behavior detection 3
1.4 Goal of this thesis 3
1.5 Outline 4
2 State of the Art 7
2.1 Road map generation from GPS-trajectories 7
2.1.1 OpenStreetMap(OSM) project 7
2.1.2 Related works of road map generation from GPS-trajectories 9
2.2 State of the art of behavior detection from trajectories 12
2.2.1 Travel mode segmentation and recognition of GPS-trajectory 12
2.2.2 Anomalous behavior detection from trajectories 13
3 Methodology Background 17
3.1 Support Vector Machines 17
3.1.1 An introduction to SVMs 18
3.1.2 Kernel Parameters and Model Selection 19
3.2 Bayesian Networks and Recursive Bayesian Estimation 21
3.2.1 Bayesian Networks 21
3.2.2 Dynamic Bayesian Networks 23
3.2.3 Recursive Bayesian Estimation 23
4 Travel mode Segmentation and Classification of GPS-Trajectories 27
4.1 Preliminaries 28
4.2 Segmentation of trajectories 29
4.2.1 Basic attributes 29
4.2.2 Segmentation of trajectories into movements 30
4.3 Teavel mode classification of movements 32
4.3.1 Patterns of travel mode movements 32
4.3.2 Travel mode classification of movements using Fuzzy-logic method 33
4.4 Constructing sub-trajectories of individual travel modes 35
4.5 Travel mode classification of sub-trajectories using SVMs method 36
4.5.1 Features for SVMs classification method 36

4.5.2 Training and testing 38
4.6 Discussion 39
5 Integration of GPS-Trajectories and an Existing Road Map 41
5.1 Preprocessing 42
5.2 Extraction of road centerlines 43
5.2.1 Matching using geometrical method 44
5.2.2 Matching using fuzzy c-means method 46
5.2.3 Construction of new centerlines 47
5.3 Estimation of number of lanes 48
5.4 Discussion 49
6 Behavior Detection from GPS-Trajectory Data 51
6.1 Detection of anomalous behavior from GPS-trajectory data 51
6.1.1 Anomalous behavior and features 52
6.1.2 The extended Markov chain 55
6.1.3 The recursive belief filter 56
6.2 Detection of collective behavior patterns 59
6.2.1 Collective travel mode change behavior 59
6.2.2 Collective anomalous behavior 61
6.3 Discussion 61
7 Experiments 63
7.1 Datasets 63
7.1.1 Tracer trajectories collected using Smartphones 63
7.1.2 OSM trajectory dataset 65
7.1.3 Chicago trajectory dataset 65
7.1.4 The use of datasets for experiments 66
7.2 Experimental results of travel mode classification 67
7.2.1 Tracer dataset 67
7.2.2 Chicago trajectory dataset 69
7.2.3 Discussion 70
7.3 Experimental results of road map refinement 71
7.3.1 OSM trajectory dataset 71
7.3.2 Chicago trajectory dataset 75
7.3.3 Tracer trajectory dataset 77
7.3.4 Discussion 78
7.4 Experiments on anomalous behavior detection 79

7.4.1 Results of anomalous driving behaviour detection 79
7.4.2 Analysis of collective behavior 82
7.4.3 Discussion 84
8 Conclusions and Future Works 85
8.1 Conclusions 85
8.2 Future Works 86
Bibliography 88
List of Figures 93
List of Tables 97
Acknowledgments 98
Curriculum Vitae 99

Chapter 1

Introduction

1.1 Motivation

A spatial trajectory is a trace generated by a moving spatial object in space. It is usually represented by an array of structured positions in space, each has a signature of specific location (geospatial coordinate tags) in time (time stamp tags); such that it is represented with 3D primitives: p = (x, y, t). The datum holds information about the representation of spatial phenomena, such as the geometry of the indoor and outdoor environment. Moreover, it also provides information about the spatio-temporal behavior of the moving objects. The huge volume of the data, e.g., spatial trajectories, can formulate and be used to analyze patterns and extract useful activity information of moving objects for a variety of location- and time- based applications, such as traffic analysis, update of existing infrastructure, anomalous behavior detection – to name a few. As a consequence, knowledge discovery from the data has become an important problem and is in increasing demand to understand the underlying nature of the data, and it aids in various decision-making processes. This thesis contributes to the general objective of knowledge discovery from crowd-sourced GPS trajectory data.

The process of mapping the earth was until recently the domain of highly skilled, well equipped, and organized individuals and groups, such as surveyors, cartographers and geographers. However, this has been altered in the last ten years, since the removal of selective availability of the GPS signal is performed on 1 May 2000. And it enabled the significant improvement of positional accuracy for simple, low-cost GPS devices. In practice, as opposed to around 100 meters before the "turn off", that enabled it to acquire the position of the GPS receiver with a positional accuracy of 6 up to 10 meters under normal conditions (HAKLAY, 2008).

GPS also enabled the development of inexpensive receivers with good positional accuracy, which helped more people than ever before collect information about different locations. GPS-data are also largely collected through mobile handheld devices. As a result, roads, paths and routable traces derived by GPS measurements are collected straightforwardly by pedestrians, public transportation commuters, bicycle riders, car drivers, and more. An updating process of topographic or vehicular data might use the spatial position derived by such measurements to enhance existing quality-inferior and out-dated road maps; other location-based services could also benefit from such data.

GPS-trajectories can track actual time and coordinates of pedestrians and regular vehicles going their usual business, and it is possible to scale to the entire area with an accuracy of 6 to 10 meters in normal condition. All the above make it possible to generate a road map from the data, and compared with the traditional way of road map generation it has the following benefits: low cost and particularly it can keep up with changes. Construction of a road map directly from GPS-trajectory data has drawn a lot of attention in recent years. The goal is to reconstruct the road centerlines as well as the mining of attribute information, for example number of lanes, road restrictions, etc. from the noisy GPS data.

GPS-trajectories can also record the spatio-temporal motion path of the moving objects. Thus, the particular behavior of moving objects can also be stored and represented by their trajectories. From an object’s trajectory, it is possible to identify different behaviors through space and time, which help us to analyze when and where something happened or what was the cause of certain actions.

One basic travelling behavior stored in a GPS-trajectory is the travel mode of the commuter collecting it. The navigational behavior of travel modes and travel mode changes is informative for variants application domains such as urban traffic. Owing to the fact that the integration of GPS-trajectory data with a road map requires the precise matching of corresponding entities, identifying correctly the road type from which the GPS-trajectory was collected (e.g. travel modes) is important for the implementation of such processes.

Another important aspect of travelling patterns that can be detected from the GPS-trajectory data is anomalous driving behavior. Anomalous patterns detection refers to the problem of finding the part of the trajectory that showing the high possibility of not conforming to expected behavior. The anomalous behavior detection from trajectories of moving objects is of great interest for various applications in the areas of navigation/driver assistance system, surveillance and emergency management.

1.2 Road map refinement from GPS-trajectory data

We consider the GPS trajectories as measurements which represent “digitization” of the true roads. Thus, the existing road data can be improved with incoming GPS-trajectories. Although the accuracy of the trajectory is not too high, due to the high number of measurements an improvement of the quality of the road information can be achieved.

In this thesis, we aim to integrate GPS-trajectories with OpenStreetMap (OSM) road data. OSM allows users to contribute GPS-trajectories and edit maps according to them. It is expected that people collaboratively manually edit the geographical information and meanwhile rectify the mistakes in the data once they spot them. In our work, we aim to eliminate the manual step and generate roads from GPS trajectory data automatically, and get a more accurate, detailed and up-to-date road map. Besides road centerlines, attribute information e.g. number of lanes are also derived through mining GPS-trajectory data.

Integrating massive crowd-sourced GPS-trajectory data with OSM road map relies on the precise matching between trajectories and road segments. The precise matching has to tackle the error both in the GPS data and the OSM data. The high degree of noise resulting from the low quality of the GPS measurements makes it on the one hand difficult to discern and separate nearby roads and on the other hand also to reconstruct the underlying structure in the road geometry, e.g. the number of lanes.

Different types of roads are often located closely to each other and trajectories that are collected from them often overlap and difficult to be separated and assigned to correct roads. Thus, besides the geometrical and statistical clustering method, identifying correctly the road type from which the GPS-trajectory is collected is important for the implementation of matching it with the corresponding road in a reference road map. The errors in the reference road map, such as missing roads, incorrect roads, etc., may also affect the integration process.

1.3 Behavior detection

Although a trajectory records the spatio-temporal motion path of an object, the underlying behavior of the object cannot be directly measured from the raw trajectory data. A number of diverse movement parameters/features (speed, acceleration, or heading changes) derived from trajectories have been used to assess the key characteristics representing the actions of objects. The identification of different types of behavior depends on the appropriate selection and extraction of movement parameters/features. Moreover, sophisticated machine learning approaches are needed to process these movement parameters/features to improve characterizing different behavioral states.

As for travel modes classification, for example, buses and cars are specifically distinct on the fact that buses have regular stops and cars do not. If the advantage of regular stops is not taken into account, bus and car travel modes will show similar movement patterns, and therefore are hard to separate. Thus movement parameters extracted from trajectories must hold the distinct and relevant characteristics of each travel modes. Otherwise, the machine learning approach would not be able to precisely classify one travel mode from others.

Anomalous driving behaviour detection from GPS trajectories aim at finding where the driver is encountering navigation problems, i.e., performing a detour, finding parking places, tending to lose the way, temporary stopover, to name a few. Unlike normal driving, when the vehicle is performing an anomalous behavior, frequently turning, detouring and coming back to a previous road may happen to a large degree. Thus the extracted movement features must reflect the above characteristics.

1.4 Goal of this thesis

In this research, we aim at developing a framework which automatically recognizes travel modes of GPS-trajectory data, and match them to corresponding roads for map refinement. We also propose a dynamic inference process to detect the anomalous behavior from individual trajectories.

The goal of this thesis is to answer the following research questions:

- How can we find the corresponding trajectories for a road in order to reconstruct its centerline?
- Beside the geometry of the road, what attribute information can be extracted from its corresponding trajectories?
- From the raw trajectory data, how can we discover movement parameters/features to assess the key characteristics describing the behavior of objects?
- Which techniques for mining movement parameters/patterns can we apply for behavior detection, and how can we improve on them?

In addition to solving the above general questions, a number of novel contributions are presented in the thesis to address the following problems.

- We use OSM road map as a reference road map and match GPS-trajectories with corresponding roads to improve the OSM road map. Besides geometrical matching, matching according to travel modes and statistical clustering method are also applied to tackle errors in GPS data and the reference map. We also extract additional attribute information from trajectories.
- A two stage classification method is developed to robustly detect travel modes of trajectories. Due to the fact that GPS trajectories are often composed of more than one travel mode, they are firstly segmented as movement segments, which are classified as pedestrian, bicycle, and motorized vehicles to find sub-trajectories that are characterized as individual travel modes. In the second stage, a breakdown classification of the motorized vehicle class as car, bus, tram and train is implemented based on sub-trajectories using Support Vector Machines (SVMs) method.
- We treat the driving behavior as a sequence of control steps rather than as a sequence of raw positions and velocities. The control steps are extracted from the trajectory of the vehicle as long-term features (turn, repetition and detour).
- To dynamically detect anomalous behavior from an individual trajectory, Markov chain is used to remodel the trajectory. And a recursive Bayesian estimator is conducted to process the Markov model and deliver an optimal probability distribution of the potential anomalous driving behavior dynamically over time.
- Collective behavior of a group of trajectories is analysed to reveal the underlying information. Group travel mode change behavior is used for the extraction of parking places associated to a specific address, while collective anomalous behavior is useful for analysing traffic conditions.

1.5 Outline

The thesis is organized as follows: Chapter 2 introduces the state of the art and selected relevant approaches on travel mode classification, map construction and behavior detection with GPS-trajectories. Chapter 3 describes the methodological background of SVMs classification and Bayesian estimation.

Chapter 4 presents the methodology of travel mode segmentation and classification. In Chapter 5, trajectories with their travel modes derived in Chapter 4 are matched to the existing road map for map refinement. In this chapter, the method of integrating GPS trajectories and the OSM road map towards a more accurate, up-to-date and detailed road map is presented.

Anomalous behavior detection from GPS-trajectories is presented and discussed in Chapter 6. In this chapter we also describe the method of collective behavior detection. The proposed methods of travel mode classification, map refinement and anomalous behaviour detection are examined with several datasets in Chapter 7. Finally, Chapter 8 summarizes the thesis and gives an outlook.

Chapter 2

State of the Art

This chapter introduces state of the art and related works of GPS-trajectory computing. In the first part, Section 2.1, an introduction to the state of the art of road generation from GPS-trajectories is conducted. This is followed by a presentation of previous work related to behavior detection with GPS-trajectories (Section 2.2). Related works conducted by researchers on the field of travel mode classification and anomaly detection from GPS-trajectories are presented and discussed.

2.1 Road map generation from GPS-trajectories

In recent years, new data sources are being more and more available, like massive amounts of GPS-trajectory data collected by volunteers while doing their own business, which in principle are manifestations of digitization of roads. The large amount of such data has made it possible to generate a road map from GPS data, and compared with the traditional way of a road map generation it has the following advantages: low cost and particularly it can keep up with changes.

A lot of projects have been set up to encourage people to contribute and make use of these valuable data. OpenStreetMap (OSM) project is one of the most extensive and successful cooperation map editing efforts. It allows its registered users to contribute trajectory data and manually edit maps according to the data.

2.1.1 OpenStreetMap(OSM) project

OSM project is founded in July 2004 at University College London (UCL) by Steve Coast. The project attempts to produce geographical information that is free to use and editable, since digital map datasets is considered to be costly and unavailable for most individuals, small businesses, and organizations (HAKLAY, 2008).

OSM follows the peer production model that is followed by Wikipedia. Registered users may contribute to the project in various ways: digitizing geographical features, uploading GPS-trajectories in GPX format from hand-held GPS devices, and correcting errors that they found in their familiar locations. Besides individual contributions, local and national authorities also make contributions to OSM by providing free geographical information. OSM also benefits from the availability of free datasets in some parts of the world.

OSM has an increasing number of volunteers to contribute to the project and has therefore gathered a large volume of GPS-trajectory data. As shown in Figure 2.1, the number of registered users is increasing dramatically in recent years. Now it is reaching 1.4 million. Simultaneously, data contribution continues to rise quickly, and up to now more than 350,000 million track points are contributed to the project by its users.

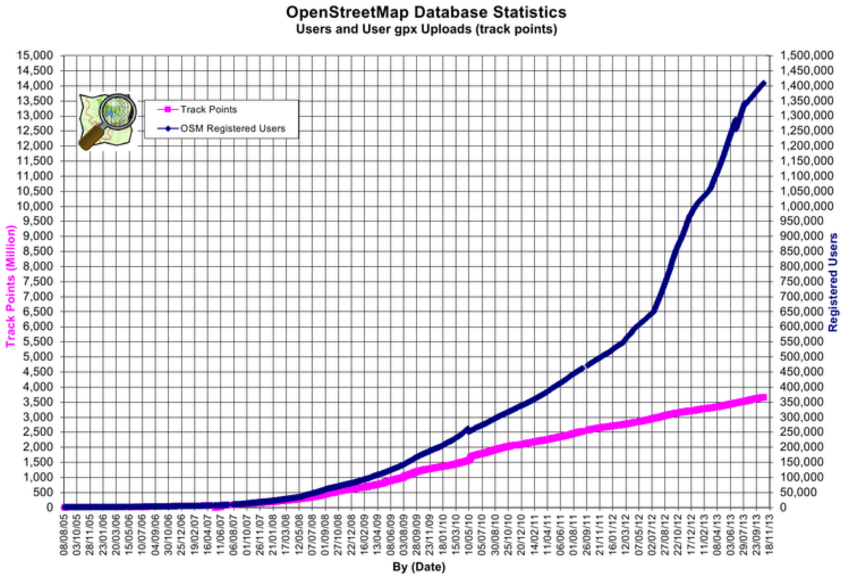
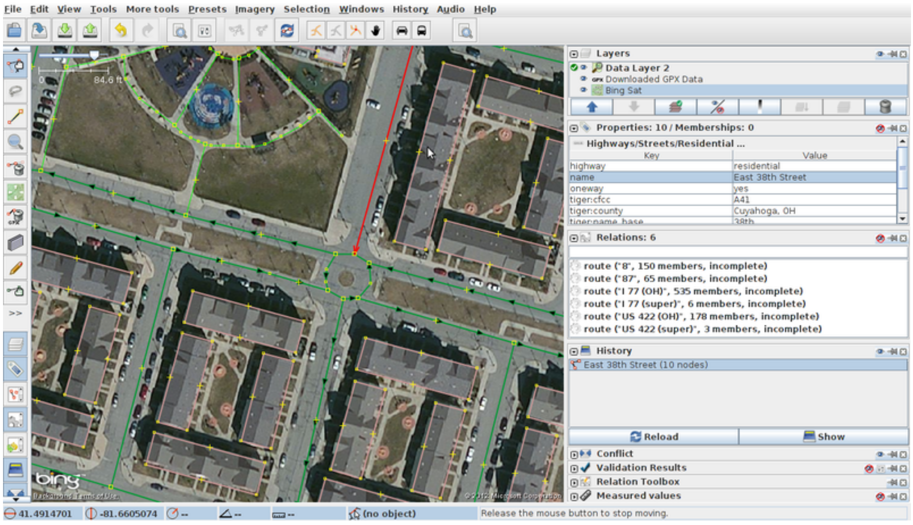


Figure 2.1: Graph of the dramatic growth in terms of registered users and contributed GPS data to OSM on a yearly basis (OPENSTREETMAP, 2014).

The users of the project are allowed to edit the map with GPS-trajectories, out of copyright maps and satellite images manually. The user contributed geographic information is obviously a core part of OSM, and OSM have developed and maintained an editing suite: Java OpenStreetMap Editor (JOSM). As depicted in Figure 2.2, JOSM is a desktop application, which lets users to import, edit, and tag, download OSM data and GPS/trajectories offline. JOSM is a feature-rich editor, and it supports users with fluid manual editing of a locally stored dataset downloaded from OSM. Moreover, OSM allows users to add and edit variant geographic attributes, such as street names, traffic restrictions.

Following the open source philosophy, OSM's technical infrastructure tries to construct own simple standard for geographical datasets other than using available standard (HAKLAY, 2008). In OSM, all entities are represented as points (node), which carry the spatial coordinates along with the attribute information, such like name, type. Linear features are defined as a list of ordered nodes, called ways. And area features are defined by closed ways.

Because of the map's free nature, it is expected that people collaboratively edit the geographical information and meanwhile rectify the mistakes in the data once they spot them. OSM doesn't have internal assurance procedure or measures of how well an area is covered.



Key	Value
highway	residential
name	East 38th Street
oneway	yes
tiger:cfcc	A41
tiger:county	Cuyahoga, OH
tiger:name_base	38th

route ("8", 150 members, incomplete)
route ("87", 65 members, incomplete)
route ("I 77 (OH)", 535 members, incomplete)
route ("I 77 (super)", 6 members, incomplete)
route ("US 422 (OH)", 178 members, incomplete)
route ("US 422 (super)", 3 members, incomplete)

East 38th Street (10 nodes)

Release the mouse button to stop moving.

Figure 2.2: The JOSM interface.

2.1.2 Related works of road map generation from GPS-trajectories

Besides OSM project, many research efforts are made to investigate the automatic map creation from GPS-trajectories. Road map generation directly from GPS-trajectory data has drawn a lot of attention in recent years. The goal is to reconstruct the road centerline as well as the attribute information, for instance number of lanes from the noisy GPS data.

Road map generation from GPS trajectories can be categorized into two types regarding whether an existing map is used. The first category is refining and updating an existing road using trajectories data, and the other category is building a road map entirely from a set of trajectories. In the following, some important works conducted in this field by researchers are described.

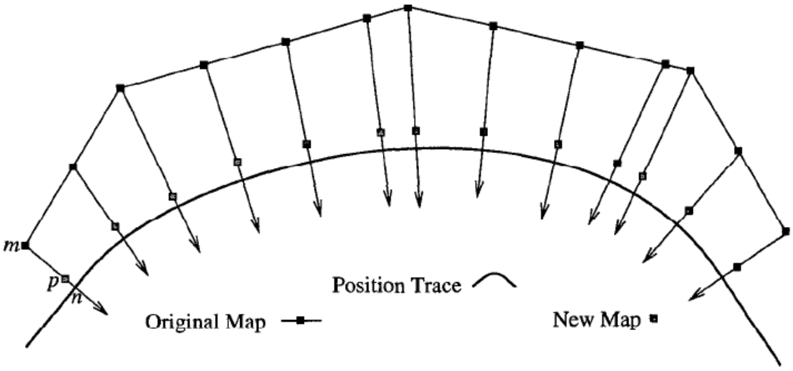


Figure 2.3: ROGERS et al. (1999) refine a road map by averaging it with a DGPS trajectory (1999).

Works in the category of refining an existing road map often involve a map matching approach, which compares the distance and heading with the prior roads. The first system that refines road map using GPS-trajectories is presented by ROGERS et al. (1999). In their work, they first refine the road map to get a more accurate centerline using the DGPS data, and then they augment road map with identification of lane structure. In the road centerline refinement procedure, as shown in Figure

2.3, they average the current centerline with an incoming trajectory, weighted by confidence in the centerline and the trajectory. The perpendicular distances from each trajectory to the new centerline are calculated and clustered into lanes.

The work presented by EDELKAMP and SCHRÖDL (2003) applies a clustering method based on the map matching result towards road map refinement. As shown by Figure 2.4, the algorithm first finds cluster seed location, which is a cluster center of a number of sample points on different traces belonging to the same road with the constraint that every trace point must be within a redefined distance and bearing difference of the cluster seed, then the seeds are aggregated into road segments and the intersection areas are defined. In their followed work (SCHRÖDL et al., 2004) they try to separate different lanes using trajectories. They propose finding clusters in perpendicular offsets of DGPS sample points from the road centerline, corresponding to the typical width of lanes.

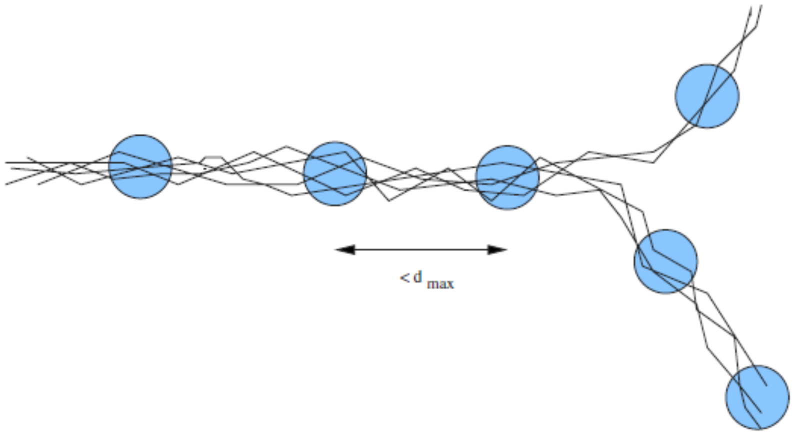


Figure 2.4: EDELKAMP and SCHRÖDL (2003) inference road map from trajectories using K-mean algorithm.

The above described map refinement methods require high quality DGPS data, which have high positional accuracy. The accurate DGPS data enable the precise map matching with the correct road rather than getting confused with the nearby roads. However, most volunteered GPS data are not as accurate as DGPS data, and the map matching becomes more complicated. Refining road map using the crowd sourced GPS trajectories have to tackle the high degree of noise in GPS data.

Other works try to infer road map entirely from trajectories. Clustering method, incremental insertion and intersection identification are normally used techniques.

CAO and KRUMM (2009) use an incremental technique based on a physical attraction model which optimizes the displacement of individual trajectories towards a modeled center line. In order to clarify the GPS/trajectories, the algorithm allows the point to move from its original position in response to forces generated on it by its neighboring points. As shown in Figure 2.5. For a point A, all segments intersecting with the orthogonal line to DE at A generate an attraction force acting on A. The strength of forces are modeled as an inverted Gaussian distribution (Figure 2.5 middle), according to which trajectory segments close to A generate stronger attractions than trajectory segments that are far away from A. A spring force corresponding to a potential well shown in Figure 2.5 (right) is applied to each point's original position to prevent the trajectories be grouped together.

Based on the clarified trajectories, they use a graph generation algorithm to infer the topology and geometry properties of the road network.

Map inference methods from GPS-trajectories based on kernel density estimation produce a binary image of the roads over the area of interest by computing the kernel density of trajectories and generate road centerlines from this binary image (DAVIES, 2006; CHEN et al., 2008).

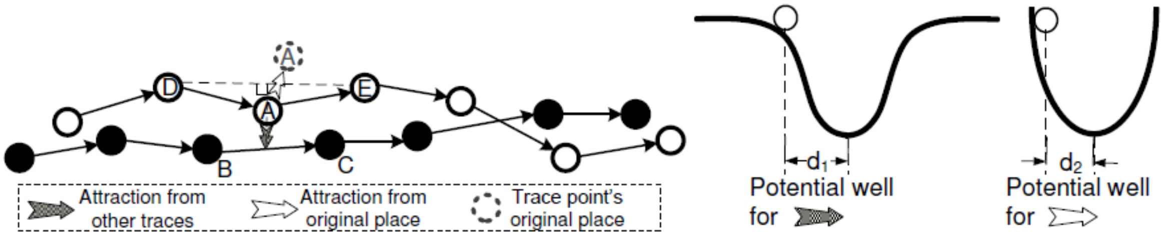

A
D
A
E
B
C
Attraction from other traces
Attraction from original place
Trace point's original place
d_1
Potential well for
d_2
Potential well for

Figure 2.5: CAO and KRUMM (2009) propose a physical attraction model, which enable the displacement of individual GPS points.

FATHI and KRUMM (2010) generate a road map by identifying and linking intersections based on GPS trajectories. A specialized shape descriptor is applied over the GPS trajectories to find places whose GPS data showing the pattern of being an intersection. Non-intersections are eliminated using a classifier that is trained on the shape descriptor. And then the intersections are connected with roads.

AHMED et al. (2004) contribute four benchmarking dataset and several evaluation methods for a standardized assessment and comparison of map construction algorithms. Sever algorithms are evaluated, which infer road map entirely from trajectories, using the benchmarking datasets.

All the above discussed map construction methods find trajectories that are collected from the same road based on their positions, and the type of the road is not taken into consideration. Thus, trajectories from different types of roads would be merged if they are closely located and parallel to each other. For example, a designed cycleway aside a car route cannot be separated. Since the crowd sourced trajectory data may be collected from commuters using various travel modes, it is possible to separate different type of roads. This on one hand enables the precise roads construction and on the other hand expands the potential application domain of the constructed map.

There exist statistical methods of inferring number of lanes using trajectories. ROGERS et al. (1999) and SCHRÖDL et al. (2004) clustering perpendicular offsets of DGPS sample points from the road centerline to find lanes. CHEN and KRUMM (2010) consider the distribution of tracks on the different lanes as a mixture of Gaussians and therefore use a Gaussian mixture model (GMM) to model the distribution of GPS traces across multiple lanes; also here prior information about lane width and corresponding uncertainty is introduced. However, the statistical assumption becomes unrealistic when dealing with the crowd source GPS data, since the high degree of errors make the trajectories cannot concentrate to form clusters.

2.2 State of the art of behavior detection from trajectories

In this section, the selected works that attempt to detect different types of behaviour in trajectories are presented. These works can be categorized into the following two groups: detecting collective behavior in groups of trajectories and identifying behavior in single trajectories.

The first group of works focus on groups of trajectories finding where common patterns can be extracted among objects with similar movements. Spatial-temporal scale problem is a major concern in identifying collective behavior in a large dataset of trajectories. Space is often divided into regions using grids or by interesting places that are detected from trajectories, while temporal scale is often implemented using time windows. GIANOTTI et al. (2007) present an approach to extract spatial-temporal sequential pattern. Sequence of spatial regions, which are frequently visited by a group of trajectories with similar transition time, are identified and used to analyse similar movement behavior. FEUERHAKE et al. (2012) incrementally identify interesting places visited by a minimal number of trajectories and characterize behavior by cluster trajectories that pass the identified interesting places. CAO et al. (2006) use the concept of time window to divide trajectories in time slices, and they investigate the collocation pattern by evaluating the trajectory in each time slice. A cross-scale approach is proposed by SOLEYMANI et al. (2014), which partition the trajectories according to spatial regions and temporal windows of different scales. Movement parameters are calculated from trajectories across different scales to explore collective behavior.

The other group of works try to detect behavior in single trajectories by analysing individual movement patterns, such as travel mode classification, interesting places extraction, intention identification and anomalous pattern detection, to name a few. Related works about travel mode classification and anomalous detection are presented and discussed in this section.

2.2.1 Travel mode segmentation and recognition of GPS-trajectory

Travel mode of a trajectory is the way of movement the commuter used when collects the data, i.e., walking, bicycle, train, tram, car, bus, etc. In theory, when compared to classic travel mode survey methods, semi-automatic and automatic classification of travel modes that is based on GPS observations, i.e., trajectories, can contribute significantly by means of accuracy and reliability. Still, since GPS observations alone supply only with positional and temporal data, specific data-mining methods are applied in order to extract the required information of travel mode type. Nonetheless, due to the fact that a single GPS-trajectory can be composed of several travel modes, most approaches include two steps: a segmentation of the trajectory into a series of possibly single travel modes; and, assigning a specific travel mode to all segments exist in the series.

A basic assumption is usually made (CHUNG and SHALABY, 2007) that walking is necessary when a mode-change occurs. This is usually characterized by low values of speed and acceleration, which are used for segmentation; this approach is sometimes referred to as change point-based segmentation method (ZHENG et al., 2008). These researches also use the time-length of each segment, assigning some thresholds for the different travel modes. Though this approach is usually found to be accurate, the research proposed here suggests using additional characterization of travel mode values and parameters, such as heading and single travel mode pattern-classifiers, thus introducing more robust and non-ambiguous segmentation for a given GPS-trajectory.

As for classification, the differentiation between five travel modes is usually made: walk, cycle, car, urban public transportation (bus and tram), and rail. Most of the existing methods compare some known preliminary travel mode related measures, e.g. rule-based values, to empirically determined values. Most commonly used values are derived from the speed and acceleration of a segment (single travel mode), such as maximum and mean speed (BOHTE and MAAT, 2009; OLIVEIRA et al., 2005; STOPHER et al., 2005). Another method suggests using particle filters that are based on learning of a Bayesian model (PATTERSON et al., 2003). Still, it was shown that these approaches might present ambiguous-classification, thus yield errors and lack the flexibility to examine proper change in pattern and uncertainty of the travel mode. Also, the determination of these thresholds is also sometimes biased from specific travel-logs (GPS-trajectories) used for analysis, i.e., the thresholds depend on a specific study-area and supplementary data. Thus, these methods are not always generic to be implemented for all environments and test-data.

To overcome the uncertainty and ambiguity existing in the data, the use of fuzzy logics as a replacement for the empirically determined values is also suggested for classification. Speed and acceleration measures are related as fuzzy sets, while fuzzy membership patterns are structured to enable travel mode classifiers via linguistic rules (TSUI and SHALABY, 2006; SCHUESSLER, 2008). Although these researches show an improvement in robustness of classification, the determination of bounds for each linguistic rules associated with each measure was found to be depended on subjective experience exist in the travel-logs. Fuzzy pattern recognition together with existing fuzzy logic classification (XU et al., 2010) showed some advantages over previous work, but still, some levels of uncertainty were remained evident.

A Decision Tree is also used (REDDY et al., 2008; ZHENG et al., 2008). In the first research the authors present its superiority to other approaches commonly used; where in the latter research, the authors show that together with a first-order Hidden Markov Model they have received promising results for classification. Still, in this case all motorized vehicles were considered as one single travel mode - as opposed to the commonly used three travel modes - and also their training data was relatively small. It also should be emphasized that the latter research used also supplementary accelerometer data for classification. This type of information is being widely used in recent researches; sometimes together with preliminary knowledge about the transportation network exists in the study-area (TROPED et al., 2008; GONG et al., 2011).

Overcoming the problems and ambiguities aforementioned, this research proposes a multi-stage classification, which introduces specific classifiers on every stage to overcome data uncertainties exist otherwise - introducing a process that is more robust. Also, it should be emphasized that six travel modes are introduced here – and not merely five – where the urban public transportation travel mode is divided to two classes: bus and tram; thus, expanding the potential of the classification process and introducing new capacity.

2.2.2 Anomalous behavior detection from trajectories

Anomalous pattern detection refers to the problem of finding patterns in data that do not conform to expected behaviour. It is of great interest for applications of navigation/driver assistance system, surveillance and emergency management. The techniques employed for anomalous pattern detection from trajectory data in the last years are summarized with the following classes:

classification based techniques, parametric or non-parametric statistical techniques, nearest neighbor based techniques, clustering based techniques, spectral techniques and information theoretic techniques.

Current researches include the work conducted by KIM et al. (2011), in which Gaussian process regression is used for the recognition of motions and activities (also anomalous events given already learned normal patterns) of objects in video sequences. PANG et al. (2011; 2013) adapt likelihood ratio test statistic to learn traffic patterns and detect anomalous behavior from taxi trajectories to monitor the emergence of unexpected behavior in the Beijing metropolitan area.

A significant amount of works related to automated anomaly detection in trajectory data involve trajectory learning, where cluster models of trajectories corresponding to normal cases, are learned from historical trajectories, and new trajectories are typically assigned an anomaly score based on the distance to the closest cluster model, or likelihood of the most probable cluster model (MORRIS, 2008).

HU et al. (2006) proposed an algorithm for automatically learning motion patterns and use these patterns for anomaly detection and behavior prediction. Trajectories are clustered firstly using both spatial and temporal similarity and then use a chain of Gaussian distributions to model each motion pattern. Based on the learned patterns, statistical methods are used to infer anomalies and predict behavior.

Besides the cluster based trajectories learning method, PICIARELLI et al. (2008) proposed a trajectory learning and anomalous behavior detection algorithm based on one-class Support Vector Machine, aiming to automatically detect and remove anomalies in the training data. They firstly evenly sample points from the raw trajectory and then model each trajectory with a fixed-dimensional feature.

BU et al. (2009) develop pruning strategies to detect anomalous behaviour based on clusters in trajectories, which regard to continuity characteristics of a group of trajectories. MA (2009) shows a method of real-time anomaly detection for users who are expected to follow normal routes. A series of axis-parallel constraints ("boxes") are generated from trajectories, and incoming trajectories are incrementally compared with a weighted trajectory.

Driver behavior detection is an interdisciplinary research field. Techniques from stochastic modeling, signal processing and machine learning, etc. are found in the literatures. Hidden Markov Models (HMMs) and Kalman filters are applied to detect driving behavior introducing the advantages of modelling both the stochastic and dynamic feature of the driver behavior.

In the work of SATHYANARAYANA et al. (2008), Hidden Markov Models is employed in both bottom-to-top and top-to-bottom approaches with three CAN-Bus features (i.e., vehicle speed, steering wheel angle and brake force) as measurements to identify three different behavior (left-turn, right-turn and lane-change).

An approach using HMMs to predict the driver behavior is presented in the work of PENTLAND and LIU (2008). They can detect driving behavior within the first 2 seconds of an action sequence with a dynamical process. MITROVIC (20059 detects driving behavior with HHMs using only vehicle speed and acceleration as raw measurement.

Kalman filter and its extension have also been proved appropriate for trajectory modelling. Recent works include (PREVOST et al., 2007), in which they present an extended Kalman filter to predict the trajectory of a moving object with the measurement data from a moving sensor -- an unmanned aerial vehicle (UAV). Kalman filter is used in (SUN et al., 2012) for the trajectory tracking based on the satellite data with weak observability and inherent large initial error.

Our work focus on finding anomalous driving behavior from an individual trajectory and a variant of recursive Bayesian filter is employed for a dynamic inference process. We treat the driving behaviour as a sequence of control steps rather than as a sequence of raw positions and velocities. The control steps are extracted from the trajectory of the vehicle as long-term features (turn, repetition and detour). These long-term features are remodelled using high-order Markov chain, which is processed by the recursive Bayesian estimator to deliver an optimal probability distribution of the potential anomalous driving behavior over time.

Chapter 3

Methodology Background

This chapter provides a background to the theory of the methodology of the thesis. In the first section, supervised learning method Support Vector Machines (SVMs), one of the main contributions of the thesis, is introduced. The model selection issue is discussed about identifying good parameters to overcome the overfitting problem. The second section introduces the theory of Bayes filters, another contribution of the thesis. Bayesian Network and recursive Bayesian estimations are described.

3.1 Support Vector Machines

Support vector machines (SVMs) are a collection of popular related supervised machine learning methods used for classification and other learning tasks used in recent years. This method projects the parameters to a high- or infinite- dimensional space and constructs a hyperplane, which can be used for classification (SMOLA and SCHÖLKOPF, 1998). SVMs are generalized linear classifiers for classification and regression, which can learn a model giving maximum predictive accuracy and meantime avoiding over fit to the training samples.

SVMs can be seen as classification and prediction tools which learn hyperplane, i.e., a linear function in a high dimensional feature space. The method introduces optimization theory to the training process that a learning bias is implemented according to statistical learning theory. SVMs are widely used for various pattern classification and regression based application domains, such as handwriting recognition, face analysis, etc. as they can produce comparable accuracy comparing with sophisticated neural networks, which are more difficult to implement.

VAPNIK (1995) first described the foundations of Support Vector Machines (SVMs). SVMs have many advantages comparing with other learning method, such as better empirical performance. The Structural Risk Minimization (SRM) theory is applied in SVMs, which has been proved to be superior to traditional Empirical Risk Minimization (ERM) theory (BURGES, 1998). The latter is used by conventional neural networks. The SRM principle enables SVM to better generalize training data, which is an important feature for statistical learning method. SVMs are not limited to solve the classification problem, and they are also able to address regression problems (VAPNIK, 1997). MEYER et al. (2003) benchmarked SVMs against other 16 classification methods, which include “conventional” methods (e.g., linear models) as well as “modern” methods (trees, splines, neural networks, etc.); by means of standard performance measures (classification error and mean squared error) SVMs presented mostly good classification performances.

3.1.1 An introduction to SVMs

In brief, as depicted by Figure 3.1, the SVMs method works as follows: it produces a model based on a set of training data (attributes together with target values), and then uses this model to predict the target value of the test data with attributes only. Given a training set of instance-labelled (x_i, y_i), where $i = 1, \dots, n$ and x□R and y□$\{1, -1\}$ the SVM finds the solution for the following optimization problem depicted in Equations 2.1 and 2.2 (CORTES and VAPNIK, 1995; HSU et al., 2003):

$$\min_{w,\xi,b}\{\frac{1}{2}W^{T}W + C\sum_{i=1}^{n}\xi_i\} \tag{3.1}$$

$$y_i(W^{T}\emptyset(x_i) + b) \geq 1 - \xi_i \;\; \xi_i \geq 0 \tag{3.2}$$

where, W is the normal vector of the hyperplane, and the parameter $b/\|W\|$determines the offset of the hyperplane from the origin along the normal vector W; ξ_i measure the degree of misclassification of the datumx_i, $C > 0$ is the penalty parameter of the error term, and, $\emptyset(x_i)$ is the function SVM projects the training vector x_i to higher dimensional space.

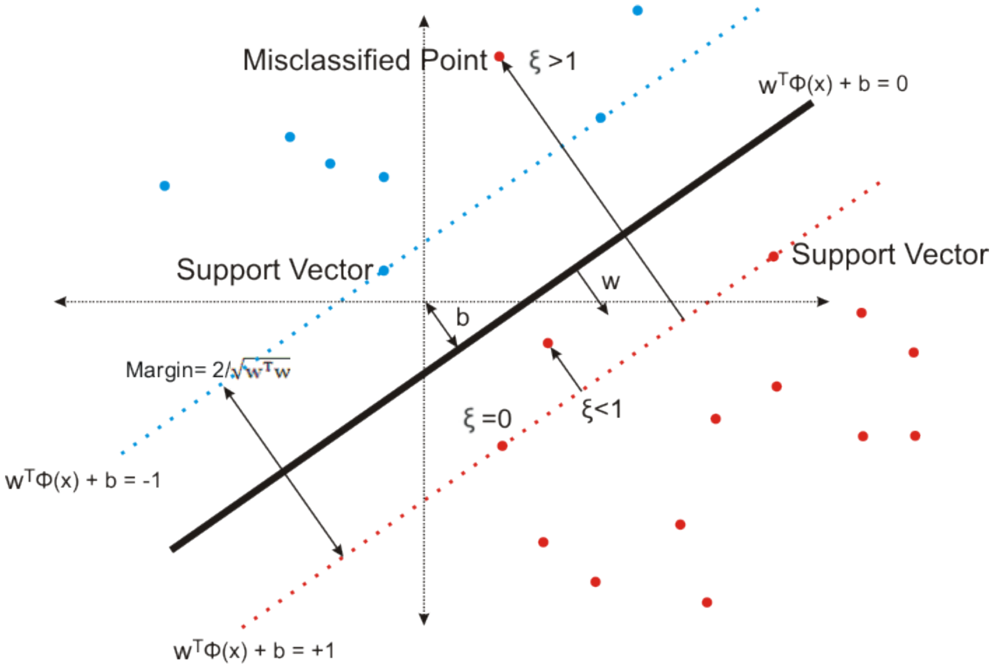


Figure 3.1: The SVM learns an optimal model which best classifies the two classes. Red dots have a label $\mathbf{y_i} = +\mathbf{1}$ while blue dots have a label $\boldsymbol{y_i} = -\mathbf{1}$.

Equation 3.3 depicts the kernel function. Gaussian Radial Basis Function (RBF) is depicted in Equation 3.4. RBF is suitable for cases where the relations between class and attributes are nonlinear and linear ones. Thus RBF kernel is the first and most popular kernel (HSU et al., 2003).

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \emptyset(\mathrm{x_i})^{\mathrm{T}} \emptyset(\mathrm{x_j}) \tag{3.3}$$

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \mathrm{e}^{(-\gamma \|\mathrm{x_i} - \mathrm{x_j}\|^2)}, \gamma > 0 \tag{3.4}$$

3.1.2 Kernel Parameters and Model Selection

In the training procedure, there are two parameters for the prediction model: C and γ, that have to be optimized. Parameter search must be done since these two parameters are unknown and the goal is to identify good parameters so that the classifier can accurately predict unknown data (i.e. testing data). However, it is not useful to identify good C and γ by simply achieving a high training accuracy, or producing a model which precisely classifies training data, in other words. The reason is that simply finding C and γ for a model that has high training accuracy may cause overfitting problem (HSU et al., 2003).

Figure 3.2depicts the overfitting taking a binary classification problem as an example (HSU et al., 2003): a large value of the regularization parameter C gives a large penalty to possible errors and overfits the classifier to training data and a smaller value of C expands the hyperplane by ignoring points close to the margin boundary. As shown in the left panel of Figure 3.2, a large penalty is assigned to margin errors, or in other words, a large value of the regularization parameter C is given to the prediction model. The two points closest to the hyperplane affect its orientation, resulting in a hyperplane that is narrow and having several other data points close to it but not included to it. Derived classifier is not good since it overfits the training data. When C is decreased (right panel of the Figure 3.2), those points are included in hyperplane; the hyperplane's orientation is optimized, producing a better margin for separating the training data. The respective classifier in the right panel does not overfit the training data and gives better cross-validation as well as testing accuracy. Thus, large value of parameter C leads to overfitting problem to the training data.

The parameter of the Gaussian kernel γ affects the flexibility of the trained classifier in fitting the training data. As shown in Figure 3.3, small values of γ enable the hyperplane boundary to be nearly linear. Large value of this complexity parameter leads to overfitting problem (HSU et al., 2003). As seen from Figure 3.3, when γ is small (top left panel in Figure 3.3) the whole set of support vectors affects the value of the discriminate function at each support vector, resulting in a smooth decision boundary. The support vector expansion increases as parameter γ is increased, resulting in greater curvature of the decision boundary. When γ is large the trained model clearly overfits the training data (bottom panels in Figure 3.3).

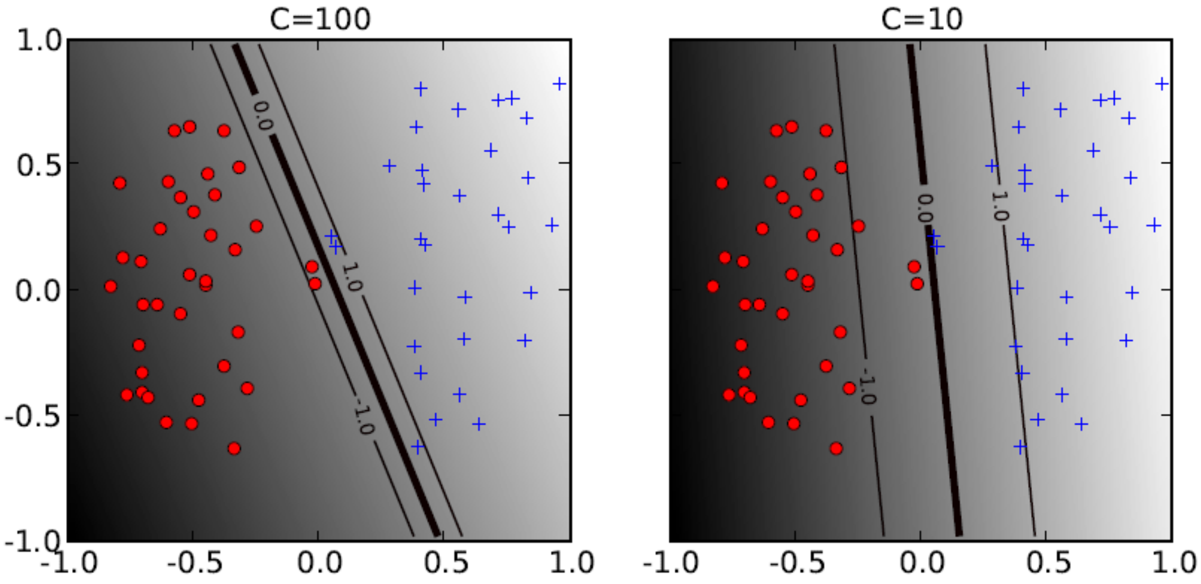


Figure 3.2: An overfitting classifier (left) and a better classifier (right) (HSU et al., 2003). The decision boundary between negative examples (red circles) and positive examples (blue crosses) is shown as a thicker line. The lighter lines are on the margin.

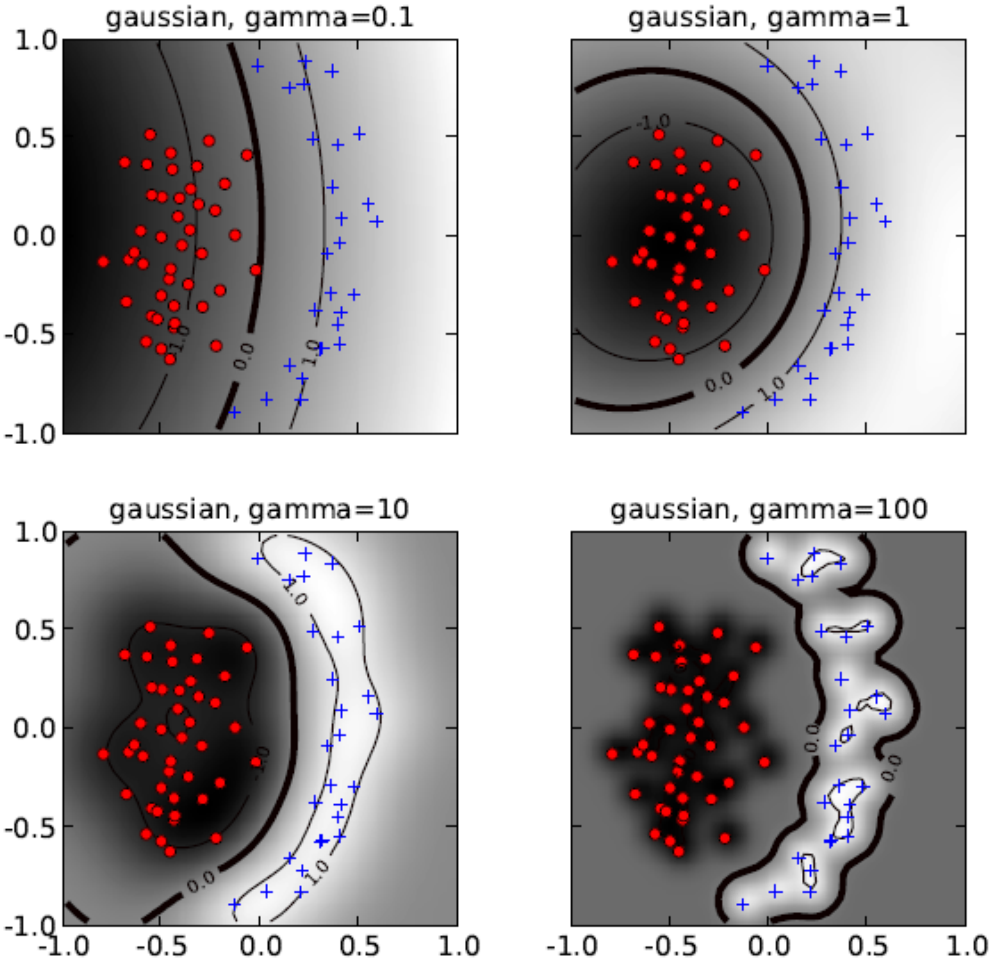


Figure 3.3: The effect of the Gaussian kernel parameter γ with a given value of $\mathbf{C}$ (HSU et al., 2003). The figure style follows that of Figure 3.2.

For Gaussian kernel there are two parameters C and γ and it is a two-dimensional searching space. Grid-search method is applied to explore this two dimensional space. The grid points are generally chosen on a logarithmic scale and for each pair of C and γ a cross-validation is done.

The cross-validation procedure can prevent the overfitting problem and identify good parameters. The k-folder cross-validation works as follows (HSU et al., 2003): dividing the training set into k subsets of equal size. Sequentially, one subset is tested using the classifier trained on the remaining k-1 subsets. Thus, each instance of the whole training set is predicted only once, and the cross-validation accuracy is the percentage of data which is correctly classified.

3.2 Bayesian Networks and Recursive Bayesian Estimation

The purpose of probabilistic inference system is to deal with the problem of rational reasoning with incomplete and uncertain information for artificial systems. Bayesian networks (BN) are a ubiquitous tool for modeling time series data including trajectory data for inference and learning problems. Many classic machine learning techniques like Hidden Markov models (HMMs), neural networks, and Kalman filters belong to the family of general Bayesian networks based estimates (MURPHY, 1998; GRIFFITHS et al., 2006). Specific types of BN models were developed to address stochastic processes, known as dynamic Bayesian networks. In this section, the basic knowledge of Bayesian Networks and the inference and learning algorithms in Bayesian Networks will be introduced.

3.2.1 Bayesian Networks

Bayesian networks, also known as belief networks, are a special case of probabilistic graphical models (GMs). Bayesian Networks are the result of the integration between the theory of probabilities and the theory of graphs. They are first introduced by PEARL (1988), and have developed as a primary theory for dealing with probabilistic and uncertain information.

A Bayesian network is a graphical way to represent conditional independencies between a set of random variables. Bayesian networks possess the ability to effectively represent and estimate the joint probability distribution over a set of random variables.

The groundwork of Bayesian networks is developed from another graphical model structure known as a directed acyclic graph. In a directed acyclic graph, random variables are represented as nodes and the direct dependence among the variables are represented as directed edges. The nodes are drawn as circles labeled by the variable names and the edges are drawn by arrows between nodes. More specifically, node A and node B are two variables in the graph and an edge from node A to node B represents a statistical dependence between them. Thus, the arrow of the edge indicates that variable B depends on variable A, or in other words variable A “influences” B. Node A is then referred to as a parent, respectively, B is noted as the child of A. The descendants of a node are its children, children’s children, and so on. A directed path from A to B is a sequence of nodes starting from A and ending in B while each node in the sequence is a parent of the following node. An undirected path from variable A to variable B is a sequence of nodes, which starting from A and ending in B with each node in the sequence is a parent or child of the following node. Figure 3.4 shows an example of a directed aryclic graph. W, X, Y, and Z are four random variables. The joint probability of the variables is as a product of conditional probabilities, that each variable potentially depends on every other variable:

$$\mathrm{P(W,X,Y,Z,)} = \mathrm{P(W)P(X|W)P(Y|W,X)P(Z|W,X,Y)} \tag{3.5}$$

However the following factorization shows conditional independence relations:

$$\mathrm{P(W,X,Y,Z,)} = \mathrm{P(W)P(X)P(Y|W)P(Z|X,Y)} \tag{3.6}$$

And, from the above factorization we can infer the conditional independence relations of the values of X and Y, W and Z.

$$\begin{aligned}\mathrm{P(W,Z|X,Y)} &= \frac{\mathrm{P(W,X,Y,Z)}}{\mathrm{P(X,Y)}}\\ &= \frac{\mathrm{P(W)P(X)P(Y|W)P(Z|X,Y)}}{\int \mathrm{P(W)P(X)P(Y|W)P(Z|X,Y)dWdZ}}\\ &= \frac{\mathrm{P(W)P(Y|W)P(Z|W,X,Y)}}{\mathrm{P(Y)}}\\ &= \mathrm{P(W|Y)P(Z|X,Y)}.\end{aligned} \tag{3.7}$$

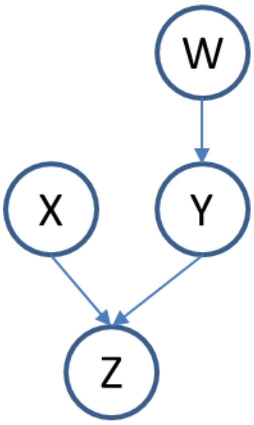


Figure 3.4: A directed aryclic graph represents the conditional independence relations in $\mathrm{P(W,X,Y,Z,)}$.

Following the above discussion, the characteristics of independence assumptions of Bayesian networks are: each variable is independent of its non descendents given its parents. More generally, two disjoint sets of variable A and variable B are conditionally independent given C, if C d-separates A and B, that is, if along every undirected path between a node in A and a node in B there is a node D such that: (1) D is both a child of the previous and the following nodes in the path, meanwhile neither D nor its descendents are in C, or (2) D is not both a child of the previous and the following nodes in the path and D is in C (GHAHRAMANI, 2001).

The graph enables to infer independence relations between variables from visual inspection other than explicitly relying on Bayesian rule. For example, W is conditionally independence from X, since Y is not both a child of W and Z, and $C = \{Y, Z\}$ is also the only path between W and X.

3.2.2 Dynamic Bayesian Networks

The framework of Bayesian Networks has been extended to Dynamic Bayesian Networks for modeling time series data and the stochastic process (DEAN and K. KANAZAWA, 1989). The assumption is the case that, an event can cause another event in the future, but not vice-versa, simplifies the structure of Bayesian networks (GHAHRAMANI, 2001). Given a graph representing the variable Y_t at time t, supposing this structure to be time-invariant and time to be discrete, the simplest models for a sequence of data $\{Y_1, \dots, Y_t\}$ is the first-order Markov model (Figure 3.5), in which each variable is directly influenced only by the previous variable:

$$P(Y_{1:t}) = P(Y_1)P(Y_2|Y_1) \dots P(Y_t|Y_{t-1}) \tag{3.8}$$

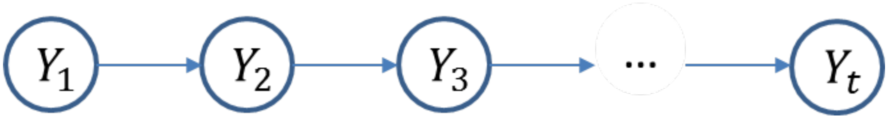


Figure 3.5: A Bayesian Network presenting a first-order Markov process.

The first order Morkov model will predict the value of Y_{t-1}only use Y_t. In order to integrate higher order interactions between variables, the first order Morkov model is extended to higher order. For example, n^{th}order Markov model allows arcs from $Y_{t-n}$to Y_t. Hidden Markov model is another way to extend Markov model, by which the observations are dependent on a hidden variable. Kalman filter, which is the continuous-state version of Hidden Morkov models, is another well-known model using the linear-Gaussian state-space model. These will be well introduced in the next section.

3.2.3 Recursive Bayesian Estimation

Bayesian Filtering, Prediction and Smoothing

Assuming variables $x_0, \dots, x_t$ are a time series of "state" variables considered on a time series from 0 to t, Variables $y_0, \dots, y_t$are a time series of "observation" variables on the same horizon. The purpose of Bayesian filtering is to compute the marginal posterior distribution or filtering distribution of the state x_k at each time step k given the history of the measurements up to the time step k (SÄRKKÄ, 2013).

The decomposition is based on three terms (DIARD et. al, 2003):

$P(x_k|x_{k-1})$ is the dynamic model or transition model which describes the stochastic dynamics of the system. It formalizes the knowledge about transitions from the state at time $k-1$ to the state at time k. The dynamic model can be a probability density, a counting measure or a combination of them depending on whether the state x_k is continuous, discrete, or hybrid.

$P(y_k|x_k)$ is the measurement model, which is the distribution of measurements given the state. It expresses what can be observed at time k when the system is in state x_k.

Finally, a prior $P(x_0)$ is defined over states at time 0.

The estimation of these models answer the following question: $P(x_{t+k}|y_{1:t})$, what is the probability distribution for state at time $t+k$ knowing the observations history from time 0 to t. When $k=0$ it is a Bayesian Filtering, which means that the present state is estimated knowing the past observations. When $k>0$, the future state is predicted from past observations. It is also possible to do “smoothing” with $(k<0)$, where a past state is updated from observations made either before or after that instant.

Hidden Markov Models

Hidden Markov models are a tool for a very popular specialization of Bayesian Filters and a tool for representing probability distributions over sequences of observations. Hidden Markov models have two defining properties (GHAHRAMANI, 2001): 1) it assumes that the observation at time t was generated by some process whose state S_t is hidden from the observer; 2) it assumes that the state of this hidden process satisfies the Markov property, which is, given the value of S_{t-1}, the current state S_t is independent of all the states prior to $t-1$. The bayesian network is drawn graphically in the form shown in Figure 3.6. Where Y_t is the observation at time t.

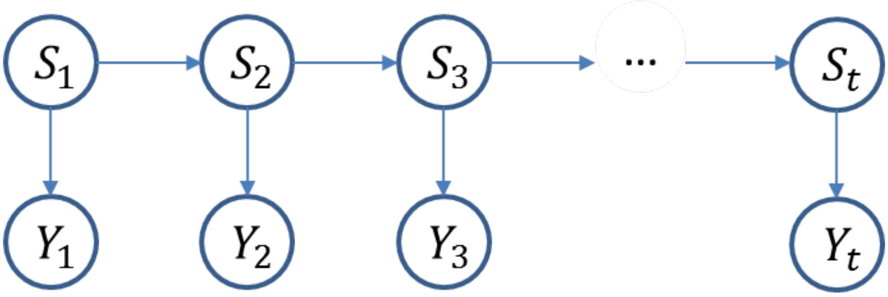


Figure 3.6: A Bayesian Network presenting the conditional independence relations of a hidden Markov model.

The joint distribution of a sequence of states and observations can be factored in the following way:

$$\mathrm{P}(S_{1:T}, Y_{1:T}) = \mathrm{P}(S_1)\mathrm{P}(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{3.9}$$

Where we have used the notation $S_{1:T}$ to mean $S_1, \ldots, S_T$. Hidden state variables are supposed to be discrete. Therefore, the transition model $P(S_t|S_{t-1})$ and the observation model $P(Y_t|S_t)$ are both specified using probability matrices.

Kalman Filters

Kalman filters are another very popular specialization of Bayesian Filters. The underlying model is a Bayesian model similar to a hidden Markov model but where the state space of the latent variables is continuous and where the transition model and measurement model are both specified using Gaussian laws with means that are linear functions of the conditioning variables (PEARL, 1988). As an optimal estimator Kalman filter infers parameters of the underlying system state from indirect, inaccurate and uncertain observations to produce a statistically optimal estimate (SÄRKKÄ, 2013). The Kalman filter minimizes the mean square error of the estimated parameters. The Kalman filter operates recursively on streams of noisy input data so that new measurements can be processed as they arrive.

Kalman filter model assumes the dynamic model and measurement models are linear Gaussian:

$$\begin{aligned} x_k &= A_{k-1}x_{k-1} + q_{k-1} \\ y_k &= H_k x_k + r_k \end{aligned} \tag{3.10}$$

Where $x_k \in R^n$ is the state, $y_k \in R^m$ is the measurement, $q_{k-1} \sim N(0, Q_{k-1})$ is the process noise, $r_k \sim N(0, R_k)$ is the measurement noise, and the prior distribution is Gaussian $x_0 \sim N(m_0, P_0)$. The matrix $A_{k-1}$is the transition matrix of the dynamic model and H_k is the measurement model matrix.

The Kalman filter is a recursive estimator. This means it uses only the previous time step and the current measurement to compute the prediction for the current state. The predict phase produces an estimate of the current state, which is also known as the a prior state estimate, using the state estimate from the previous time step. Although it is an estimate of the state at the current time step, it does not take observation information from the current time step into consideration. Once the next measurement is observed, in the update phase, the prior prediction is updated with current observation information to adjust the state estimate, which is termed the posterior state estimate (SÄRKKÄ, 2013).

Typically, the prediction and updating phases alternate. The predicted state is kept until the next observed measurements, and then the update integrates the observation. However, this is not always the case; if an observation is unavailable for some reason, the update phase may be skipped and prediction steps may be performed without updating.

Extensions and generalizations of Kalman filters are developed to deal with nonlinear systems using a first order Taylor's expansion to consider that these models are locally linear. This generalization is commonly called the Extended Kalman Filter.

The Kalman filter method is popular because that it is an optimal estimator and it can produce good results in practice due to its optimality and structure. It is recursive by nature and is therefore convenient for online real time processing. Finally it is easy and efficient to formulate and implement in practical use.

Chapter 4

Travel mode Segmentation and Classification of GPS-Trajectories

Six travel modes are taken into consideration, which supposedly consists of different movement-patterns: walk, bicycle, car, bus, tram and train. The assumption is that every GPS-trajectory stores some unique and relevant characteristics that are derived from a specific travel mode resultant by the road-type it was acquired on.

Most common travel, or traffic, characteristics used in related research nowadays are speed and acceleration. Still, these two unique characteristics might not always be sufficient, as ambiguities (different travel modes might have similar characteristics) and errors are also propagated onto the travel mode trajectory. The use of supplementary parameters extracted from raw GPS data is introduced, such as stops, heading changes and travel time, etc., to achieve more reliable classification results.

A problem also arises when a single trajectory is composed from several sub-trajectories, each corresponding to a different travel mode. Thus, the strategy is to first segment the trajectory to derive sub-trajectories of individual travel modes.

A multi-stage classification method is applied to identify each travel mode. The motivation for adopting a multi-stage method is that the three classes: walk, bicycle and motorized vehicles, consist of unique characteristics, which are essential for constructing sub-trajectories of individual travel modes. Furthermore, under normal conditions, a period of walking or stop is necessary in order to transfer from one travel mode to another among car, tram or train travel modes. Thus, two or more travel modes of motorized vehicles are not likely to be put in a single sub-trajectory. Sub-trajectories of individual travel modes then contribute to the second stage classification of motorized vehicles using SVMs method.

4.1 Preliminaries

A trajectory records the path a moving object follows through the space using a discrete sample of time-stamped locations. We define the trajectory as a mapping from a sequence of positions, or vertices, $p_0, p_1, \dots, p_n$ for time stamps $t_0, t_1, \dots, t_n$ to the plane, where p_i is the location of the moving object at time t_i. We denote the links between any two adjacent points p_{i-1}, p_i for $1 \le i \le n$ as edges e_i of the trajectory. For any time range $[t_i,\ t_j] \in \{t_0, \dots, t_n\}$ with $t_i < t_j$, we denote the segment of trajectory T from $t_i$to time t_j by $T[p_i, p_j]$. These are depicted in Figure 4.1.

Sub-trajectories of a trajectory are a partition of the trajectory into segments with the criteria that travel modes changes exist. Thus a sub-trajectory $T[p_t, p_{t'}]$ is a segment of a trajectory $T[p_0, p_n]$ inside which any location point p_i for $0 \le t \le i \le t' \le n$ is collected from the same individual travel mode. Meanwhile, points p_{t-1} and $p_{t'+1}$are collected from other travel modes. A sub-trajectory is composed of a collection of movement segments. Figure 4.1 depicts an example trajectory, which is composed of three separate travel mode sub-trajectories: train, walk and bus.

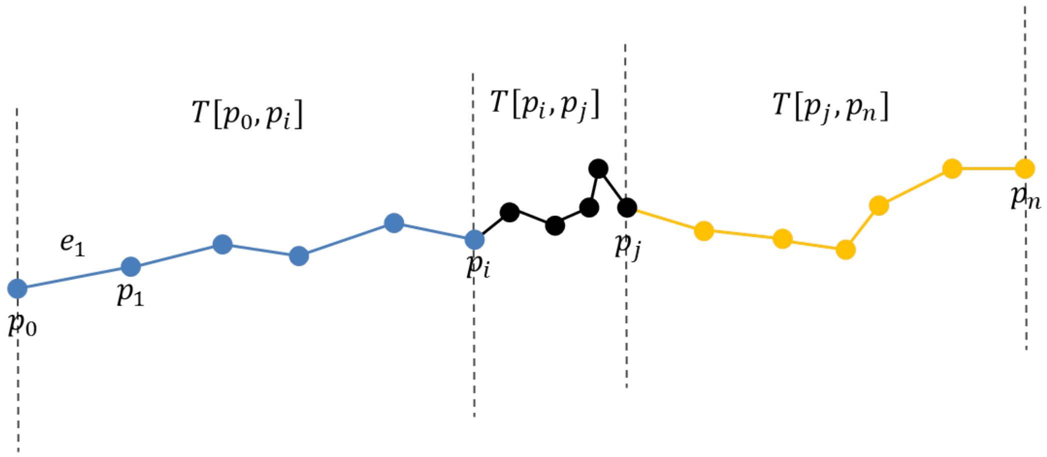


Figure 4.1: An example of a trajectory that is composed of 3 sub-trajectories: train (blue), walk (black) and bus (yellow).

We define a movement $T[p_t, p_{t'}]$ for $0 \le t < t' \le n$ as a segment of the trajectory T, that is divided by two stops. In other words, for $[t, t']$ the object completes a movement that it starts to move, speeds up, slows down and eventually stops.

In the first stage, the segmentation and classification of travel modes are calculated using speed related attributes and heading related attributes, which are computed using the known locations and time stamps. The attributes are provided for vertices and are constant on every edge. In the second stage, the attribute for SVM classifier is calculated for each movement segment comprising a sub-trajectory.

4.2 Segmentation of trajectories

4.2.1 Basic attributes

The basic attributes that are used for segmentation and classification are speed, and heading. Further relevant attributes, such as accelerations, heading changes and stops are computed from these basic attributes.

Speed

The speed for any edge e_i with $1 \leq i \leq n$ of the trajectory $T[p_0, p_n]$ is assumed to be constant. At any vertex p_{i-1}, the speed is the same as the speed on the edge e_i, that is the edge after the vertex. The speed for edge e_i is calculated from the distance of the vertices p_{i-1}, p_i and the time interval between them.

The positional accuracy of the GPS signal can reach several meters under normal conditions (WOLF, 2006). However, in some situations, such as lack of sufficient satellites, equipment not being ideally positioned, signal being reflected by tall buildings or bad weather, the positional accuracy can be worsened. The errors are reflected on the position of the acquired GPS data. Thus, a mean filtering technique is applied towards smoothing and preliminary reduction of error effect of the calculation of speed attributes.

The mean filter performs like a sliding window covering n temporally adjacent values of speed of z_i. For a measured point p_i, the estimate of the unknown true value z'_i is the mean of z_i and its $n-1$ predecessors in time. Thus its speed errors are reduced by averaging its neighborhoods. The equation form of the mean filter is as following:

$$z'_i = \left(\sum_{i-n+1}^{i} z_i\right)/n \qquad (4.1)$$

Here n is 5, that means the range of the filtering is five travel-epochs under common conditions.

Heading

The heading for the moving object at any time is the direction in which it is travelling. We give the heading to any vertex p_i as the same as the direction of edge e_{i+1}, which is the edge after the vertex.

We use north-based azimuth system to calculate the heading for each edge. North is defined to be the zero, and the angle is measured clockwise from the north. For examples, northeast has azimuth 45° and east 90°.

Heading smoothing is not implemented here, because heading is not continuous by nature. Thus smoothing might remove its characteristic, and degrade its reliability as a travel mode parameter required by classification method.

4.2.2 Segmentation of trajectories into movements

Individual movement segments are derived by identifying stops. Identification of stops also enables us to filter-out the stops-data that should not be analyzed when calculating movement parameters required by SVM classification, such as average speed of a movement.

We define a stop as a trajectory segment $S[p_i, p_j]$ of $T[p_0, p_n]$where $[t_i,\ t_j] \in \{t_0, \dots, t_n\}$ with $t_i < t_j$, in which the object move at a very low speed and its position is changed in the time range $[t_i,\ t_j]$ with a very small distance. Here, stops are not just observations with low speed, but a sequence of observations that last for a period of time. This is depicted in Figure 4.2 (top) as black segments, that have very low speed and very small distance changes.

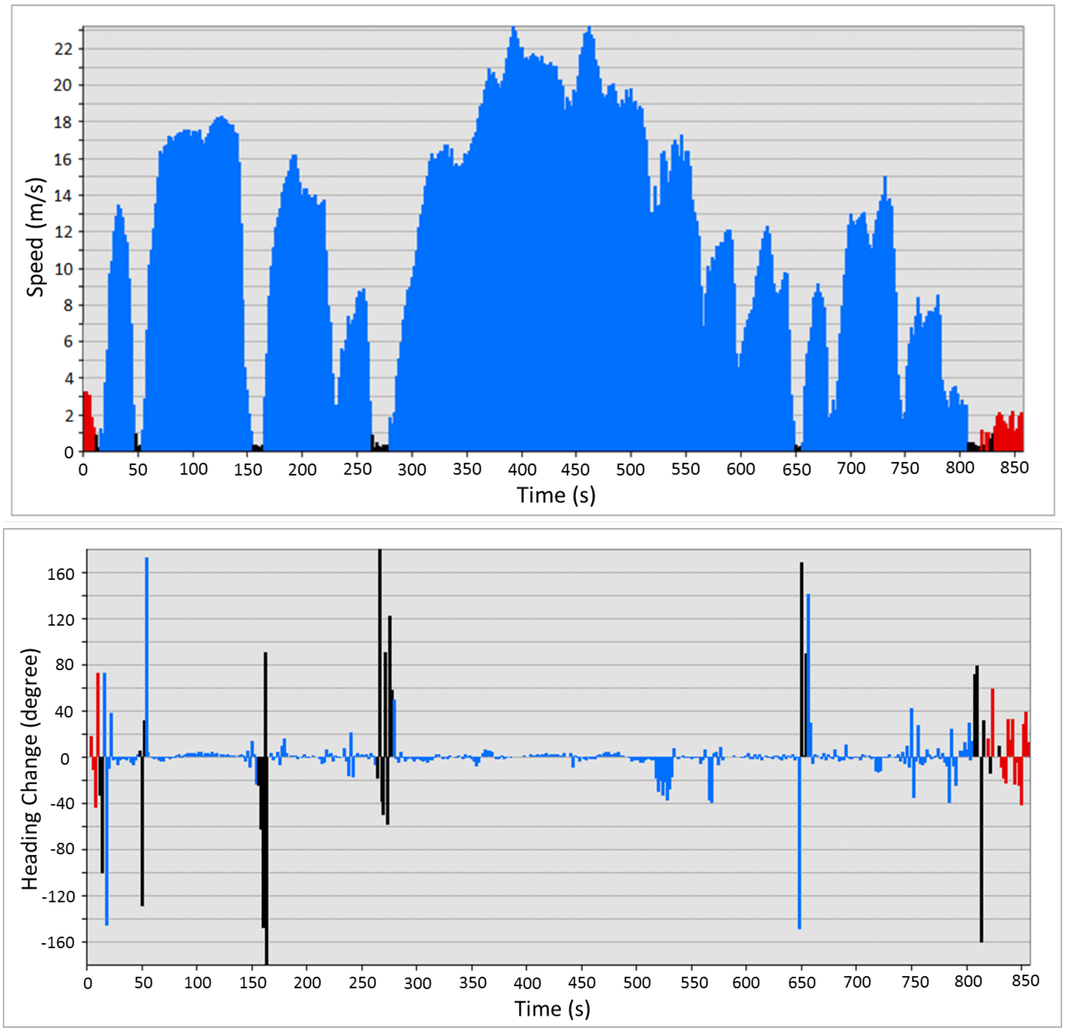


Figure 4.2: Speed pattern of movement segments and stops (top) and corresponding heading changes magnitude (bottom). GPS-trajectory representing approximately 15 minutes of travelling is divided into different individual movement segments. Car movement segments are in blue, walk segments are in red, and identified stops in black.

Beside the commonly used values associated with stops, namely small distance changes per-time and low speed value, the use of magnitude in heading change is introduced; this parameter was found to be vital for a robust identification of stops. As illustrated in Figure 4.2 (bottom), stops are always accompanied with large magnitude values in heading changes, which cannot be explained by realistic movement changes. The large values result when the object travels with very low speed values or stops, thus the large and random magnitude values in heading changes are due to relatively small changes in position. Thus, the following criteria are used to detect the stop segments exist in the GPS-trajectory.

GETstop rules:

- Small change in position: where distance change for 5 consecutive seconds is less than 5 meters, AND
- Small speed values: speed value is less than 1 m/s, OR
- Large magnitude in heading changes: change of heading is larger than 100 decimal degrees.

The algorithm takes a greedy strategy in finding stop segments that it makes the stop segments as long as possible to avoid small and useless segments. Let s be the first point that satisfies the criteria of being a stop, we find the longest segment by finding a vertex p_j so that the vertex p_{j-1}belongs to the stop segment but p_j does not. The incremental method is applied for the implementation of the strategy. The algorithm fist calls 5 point neighborhood $T[p_s, p_{s+5}]$ and if the 5 consecutive points satisfies the criteria, let the next point after the neighborhood be the start point and incrementally call $T[p_{s+6}, p_{s+11}]$, $T[p_{s+12}, p_{s+17}]$, ..., until the test of the criteria fails at p_{j-1}. Thus, the stop segment $T[p_s, p_{j-1}]$ is found.

The “GETstop” algorithm works as follows:

- From the first point p_s on (if it is the beginning of the trajectory, then $s = 0$), if the distance from that point to its fifth consecutive neighbour p_{s+5} is less than 5 meters, go to step 2.
- Check each point of the 5 points $T[p_s, p_{s+5}]$: if its speed is smaller than 1 m/s or change of heading magnitude is larger than 100 degrees, mark the point as stop and check the next point. Else, break the trajectory from that point, and go to step 1. If no break occurs, go to step 3.
- Let the vertex p_{s+6}be the beginning point, do step 1. Stop when the end of trajectory is reached.

Figure 4.3 is an example of a trajectory after identifying stops. The result of this process is that stops are labeled and two adjacent movements are separated by stops. The correctly found stops are important for the identification of movement segments, and further aid the calculation of classifier attributes by separating stops from travelling segments, such as average speed of a movement, stop time, etc.

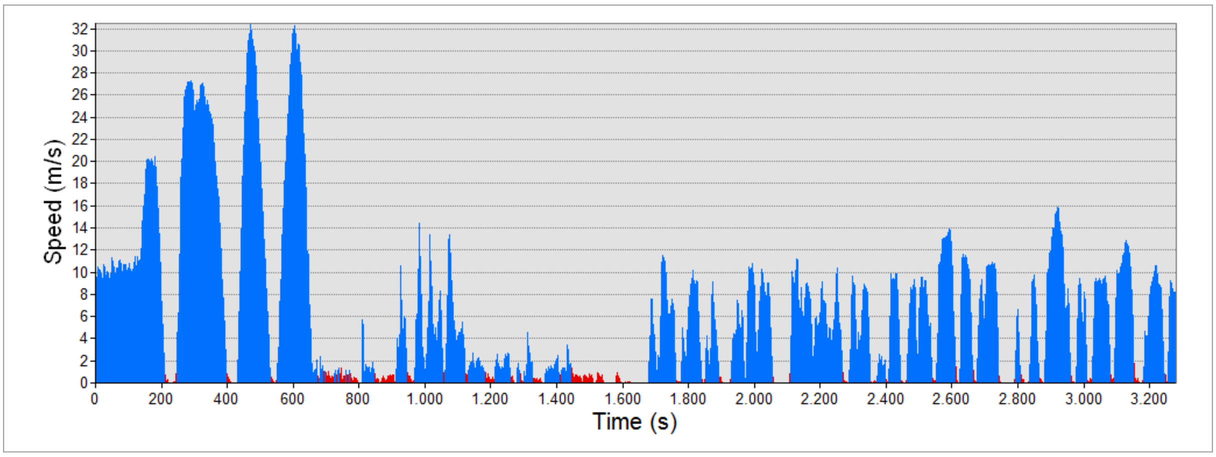


Figure 4.3: Speed pattern of an example trajectory with stops between two adjacent movements are identified. Movement segments are presented in blue and stops are in red.

4.3 Teavel mode classification of movements

4.3.1 Patterns of travel mode movements

As mentioned before, a GPS trajectory is not necessarily derived from a single travel mode; instead, it is often composed of several different travel modes. For example in Figure 4.3, the trajectory is a combination of three travel modes: train and walk and bus.

Before any classification is carried out, a separation of the trajectory into segments of an individual travel modes has to be implemented, which are characterized as movement segments. A movement segment is composed of a segment separated by two stops as depicted in Figure 4.3. After all movement segments composing a single GPS-trajectory are identified (cf. 4.2.2), the classification is applied on these movement segments, finalized by linking neighboring segments that have been classified with the same travel mode to form a sub-trajectory.

However, after comparing the segments' speed characteristics of the six different travel modes, displayed in Figure 4.4, it was found that the characteristics of the pedestrian and bicycle travel modes are prominently different from all other travel modes, e.g., motorized vehicles. Additionally, a classification of car, bus, tram and train that is solely based on the movement segments might result in ambiguous results, as the segments have similar characteristics, and the characteristics of the whole sub-trajectory of a specific travel mode are not utilized. For example, buses and cars are specifically different on the fact that buses have regular stops and cars do not. If the advantage of stops is not taken into account, bus and car travel modes will have similar movement-patterns, and thus are hard to separate. Since no transfer between two motorized vehicles travel modes is possible to be completed without walking or stopping for a period of time, it is unlike to put travel-movements from different motorized-vehicles travel modes in one sub-trajectory.

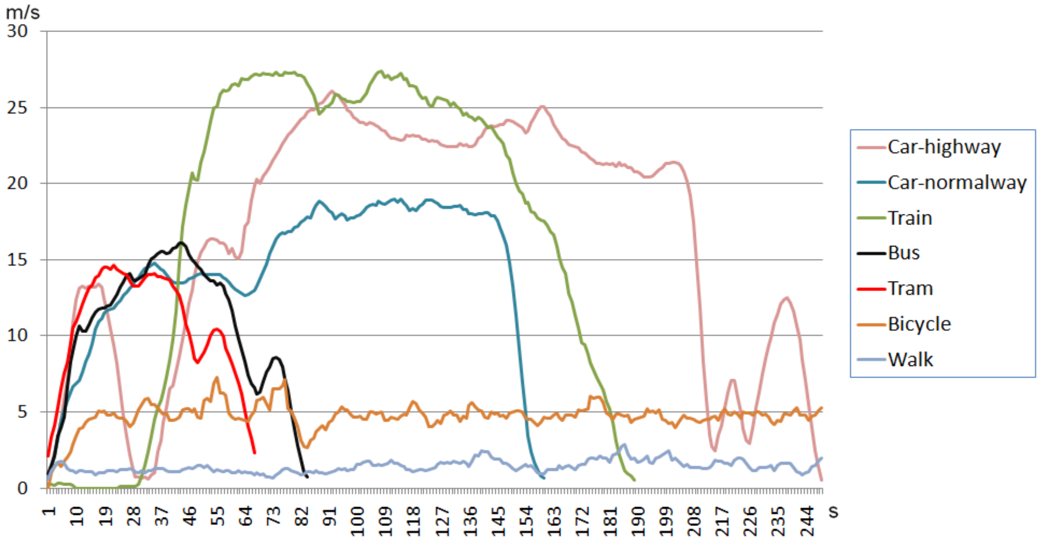


Figure 4.4: Typical speed patterns of different travel mode movement segments.

As a result, adopting a multi-stage method is employed: in the first stage, pedestrian and bicycle travel modes are differentiated from motorized vehicles based on specific characterizations and specification of their movement segments. In the second stage, segments are linked up to form sub-trajectories, and consecutively car, bus, tram and train travel modes are classified based on the specific characterizations and specification of the sub-trajectories.

4.3.2 Travel mode classification of movements using Fuzzy-logic method

The speed related characters are very important for identifying travel modes, especially for the first stage of the classification. The mean speed, maximum speed, mean acceleration and maximum acceleration are utilized. These are calculated for each individual travel-movement segment. In order to get more reliable parameters and reduce errors that might exist in using one observation alone, maximum speed and maximum acceleration are calculated on the basis of the average values of the top 5 values of the segment.

Heading related parameters, namely mean and maximum heading magnitude changes, are also used. The heading change is calculated to be in the range of (-180°, 180°). When calculating mean heading changes all values are transferred to positive because the magnitude alone is of importance. It is shown in the bottom panel of Figure 4.2, that the magnitude of heading has high correspondence with different travel modes. While maximum heading change corresponds to stops, walking always shows a large magnitude value for the average heading changes.

As depicted in Figure 4.5, three basic classifiers - mean speed, maximum speed and mean heading - are used for the first classification stage. Wide enough ranges for the three classifiers are used to include all possible segments into consideration whilst avoiding making wrong classification. Later, additional classification parameters, such as maximum heading changes, mean heading changes are used to validate correct travel mode separation. From Figure 4.5 it is clear that there are some overlay areas for the parameters used between: stop and walk, walk and bicycle, and bicycle and motorized vehicles. Taking all three under consideration, a minority of segments will fall into these overlay areas simultaneously; to solve these ambiguities, extra parameters are introduced:

- For stop and walk overlay area maximum heading change is introduced. As mentioned before, stops are always accompanied with high heading changes. If the maximum heading change of a segment is larger than 80 degrees, it is identified as a stop.
- For walking and bicycle separation, mean heading is used, for the reason that pedestrian trajectories are always accompanied with higher heading changes than bicycle ones.
- For bicycle and motorized vehicles overlay area the maximum acceleration is introduced; this is because it was found that unlike motorized vehicles when bicycle travels with a relatively high speed (mean speed > 5 m/s) there is always an evident high value of acceleration (maximum acceleration > 4 m/s^2).

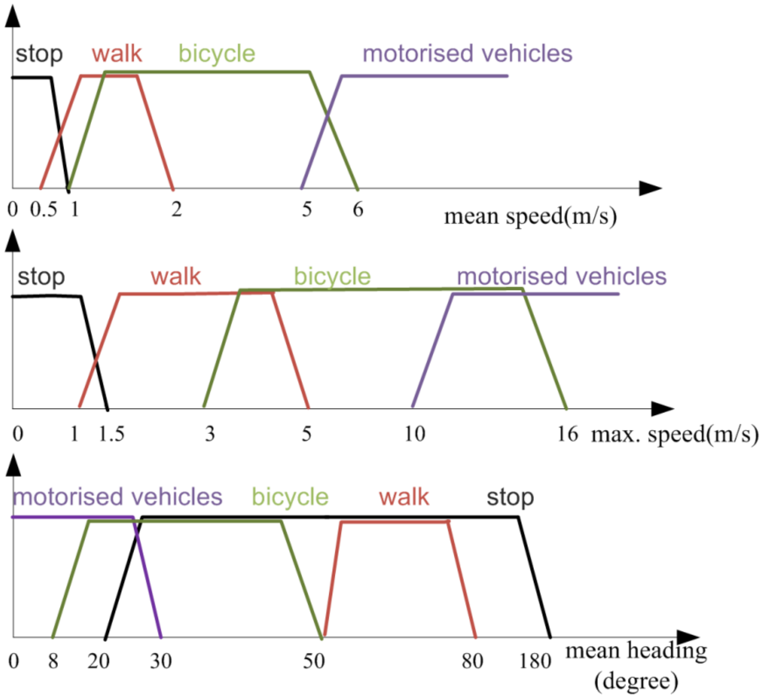


Figure 4.5: Value range of mean speed (top), maximum speed (middle), and mean heading changes (bottom) for different travel modes.

With the use of these, walk, bicycle and motorized vehicles travel modes are identified correctly. As shown in Figure 4.6, each movement segment is identified as one of the three types: walk, bicycle or motorized vehicles. In the following section, these movement segments are linked up to derive a sub-trajectory.

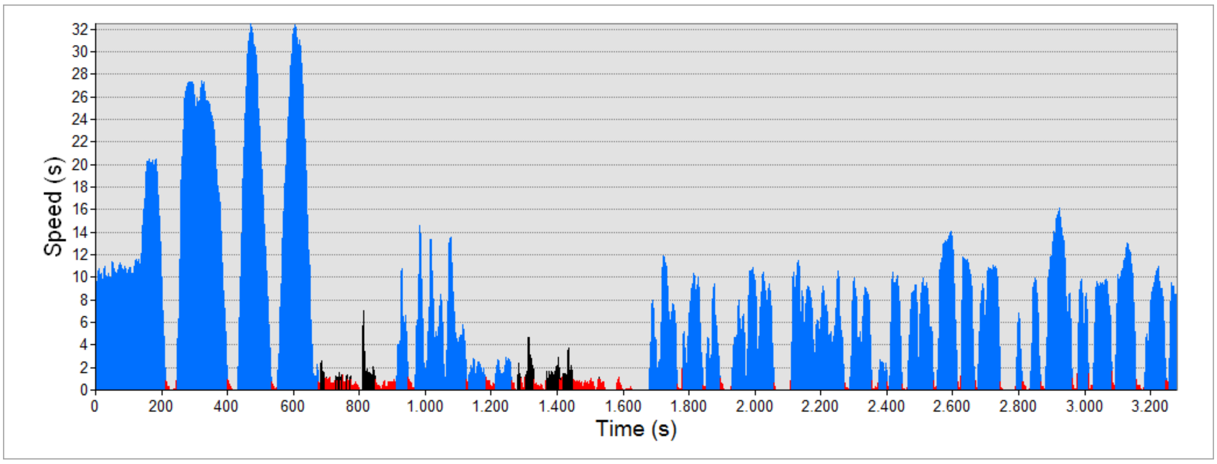


Figure 4.6: The example trajectory after performing the first stage classification. Each segment is identified as vehicle (blue) or walk (black). Stops are represented in red.

4.4 Constructing sub-trajectories of individual travel modes

Neighboring movement segments of the same travel mode are linked up to form sub-trajectories, which are assigned a travel mode of pedestrian, bicycle or motorized vehicle. The segments are checked with specific predefined rules to ensure that they are not incorrectly classified before joined to a sub-trajectory. For example, a bus segment, which has a relatively low speed, may be wrongly identified as a bicycle segment. However, if this segment's neighboring segments are both bus travel mode, and since it is not possible to transfer from bicycle to bus without stop or walk, the travel mode is corrected to bus travel mode.

In order to correct the possibly wrongly classified segments, rules are applied in the linking procedure according to the basic travel knowledge.

GETsub-trajectory rules:

- A vehicular sub-trajectory should not last less than 120 seconds; the use of 120 seconds is designed to eliminate sub-trajectories that are too short, e.g., have no significance, or are wrongly classified.
- Stop duration between two movement segments of one sub-trajectory should be less than 120 seconds; if the stop duration is longer than 120 seconds the sub-trajectory should be treated as two individual sub-trajectories.
- No directly transformation from bicycle to motorized vehicle is possible, unless at least 120 seconds of walking or stop took place; this time duration threshold of 120 seconds is used to avoid linking two different modes together.

The construction of sub-trajectory of the example trajectory is shown in Figure 4.7. 5 sub-trajectories are derived: train, walk, bus, walk and bus. 3 vehicle sub-trajectories are found, which are divided by two walk sub-trajectories.

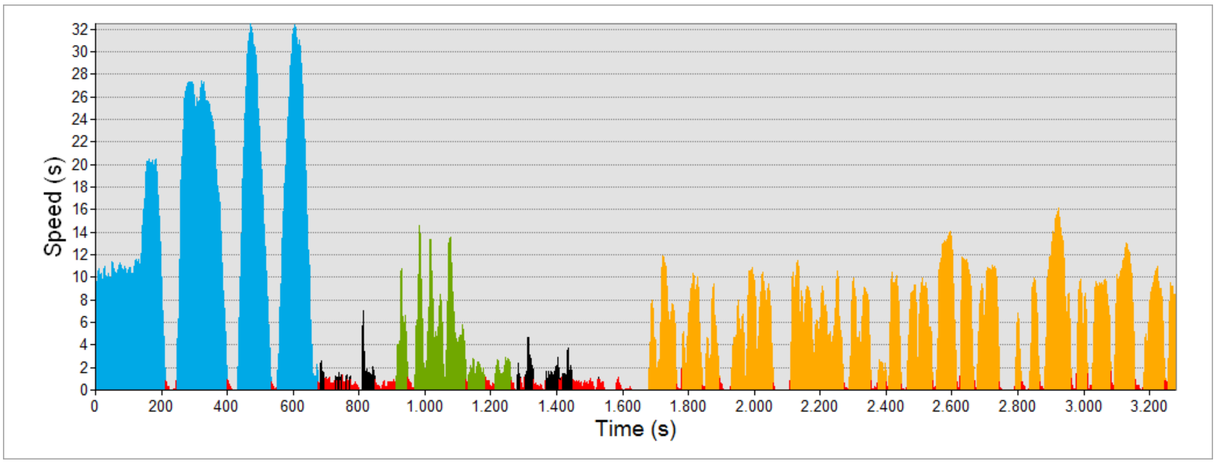


Figure 4.7: Sub-trajectory construction result of the example trajectory. 3 vehicle sub-trajectories are identified and are represented in blue, green and yellow. Walks are shown in black and stops in red.

4.5 Travel mode classification of sub-trajectories using SVMs method

After the first-stage classification, the use of the supervised learning method Support Vector Machines (SVMs) is employed to classify motorized vehicles sub-trajectories class to specific travel modes of car, bus, tram and train. A Gaussian kernel is used to transform the input attributes and train an optimal model to classify testing data.

SVMs method requires small training datasets, as it has high generalization. Another advantage of SVMs is that it provides a good out-of-sample generalization, if appropriate regularization parameter C and Gaussian kernel parameter γ are chosen. In other words, by identifying an appropriate generalization grade, SVMs can be robust even when the training sample has some degree of bias.

4.5.1 Features for SVMs classification method

The second stage of classification using SVMs is based solely on the sub-trajectories of motorized vehicles. The entire sub-trajectory is treated as a single object, and the attributes of each sub-trajectory are presumed to describe the characteristics of a unique travel mode. Buses, trams and trains - other than cars - should present in the data regular stops together with similar travel duration between two consecutive stops, while the total amount of time for each stop is supposed to be longer than that of cars. In order to extract the described characteristics of the sub-trajectory, speed, acceleration and time information are derived from each travel-movement and the average and standard deviation are calculated from these values. Suppose a sub-trajectory T is composed of n movement segments $T\{T_0, T_1, \dots, T_1\}$, we denote the maximum speed of a movement segment T_i as $V_{max}(i)$, the average speed as $V(i)$, the maximum acceleration as $A_{max}(i)$ and the travel time as $M(i)$; and we denote the stop time of the whole sub-trajectory as M_{stop}.

According to these, total of 11 attributes $\{p_0, p_1, \dots, p_{11}\}$ are used as attributes for the SVM implementation:

1) Mean of maximum speed

$$p_1 = \frac{1}{N} \sum_{i=0}^{n} V_{max}(i) \tag{4.2}$$

2) Standard deviation of maximum speed

$$p_2 = \sqrt{\frac{1}{N-1} \sum_{i=0}^{n} (V_{max}(i) - p_1)^2} \tag{4.3}$$

3) Mean of average speed

$$p_3 = \frac{1}{N} \sum_{i=0}^{n} V(i) \tag{4.4}$$

4) Standard deviation of average speed

$$p_4 = \sqrt{\frac{1}{N-1} \sum_{i=0}^{n} (V(i) - p_3)^2} \tag{4.5}$$

5) Mean of maximum acceleration

$$p_5 = \frac{1}{N} \sum_{i=0}^{n} A_{max}(i) \tag{4.6}$$

6) Standard deviation of maximum acceleration

$$p_6 = \sqrt{\frac{1}{N-1} \sum_{i=0}^{n} (A_{max}(i) - p_5)^2} \tag{4.7}$$

7) Mean of average acceleration

$$p_7 = \frac{1}{N} \sum_{i=0}^{n} A(i) \tag{4.8}$$

8) Standard deviation of average acceleration

$$p_8 = \sqrt{\frac{1}{N-1}\sum_{i=0}^{n}(A(i) - p_7)^2} \tag{4.9}$$

9) Mean of travel time

$$p_9 = \frac{1}{N}\sum_{i=0}^{n} M(i) \tag{4.10}$$

10) Standard deviation of travel time

$$p_{10} = \sqrt{\frac{1}{N-1}\sum_{i=0}^{n}(M(i) - p_9)^2} \tag{4.11}$$

11) Ratio of stop time in respect to travel time

$$p_{11} = \frac{M_{stop}}{\sum_{i=0}^{n} M(i) + M_{stop}} \tag{4.12}$$

Each travel movement segment within an individual sub-trajectory is used for the calculation of the aforementioned attributes. The attributes are scaled before applying SVMs to range (0, 1). Both the corresponding attributes of training and testing data are scaled in the same way. The main advantage of doing so is avoiding the attributes of greater value ranges dominating those in smaller numeric ranges, together with benefit of reducing calculation complexities (HSU et al., 2003).

4.5.2 Training and testing

The cross-validation procedure can prevent the overfitting problem and identify good parameters. The *5*-folder cross-validation is applied in the training phase. The training set is divided into 5 subsets of equal size. Sequentially, one subset is tested using the classifier trained on the remaining 4 subsets. Thus, each instance of the whole training set is predicted only once, and the cross-validation accuracy is the percentage of data which is correctly classified. In our implementation, we use a 5-folder cross-validation.

In the training procedure, there are two parameters for the prediction model: C and γ, that have to be optimized. Parameter search is performed since these two parameters are not available and it is important to identify good parameters so that the trained model can accurately predict unknown testing data. A grid-search is applied, and for each pair of C and γ a cross-validation is done, and the pair with the best cross-validation accuracy is selected.

After the prediction model is derived through training a set of sub-trajectory data with travel modes are labeled, the model is used to predict unknown testing data.

As shown in Figure 4.8, 3 vehicle sub-trajectories of the example trajectory are classified to train and bus respectively after applying SVMs classification method.

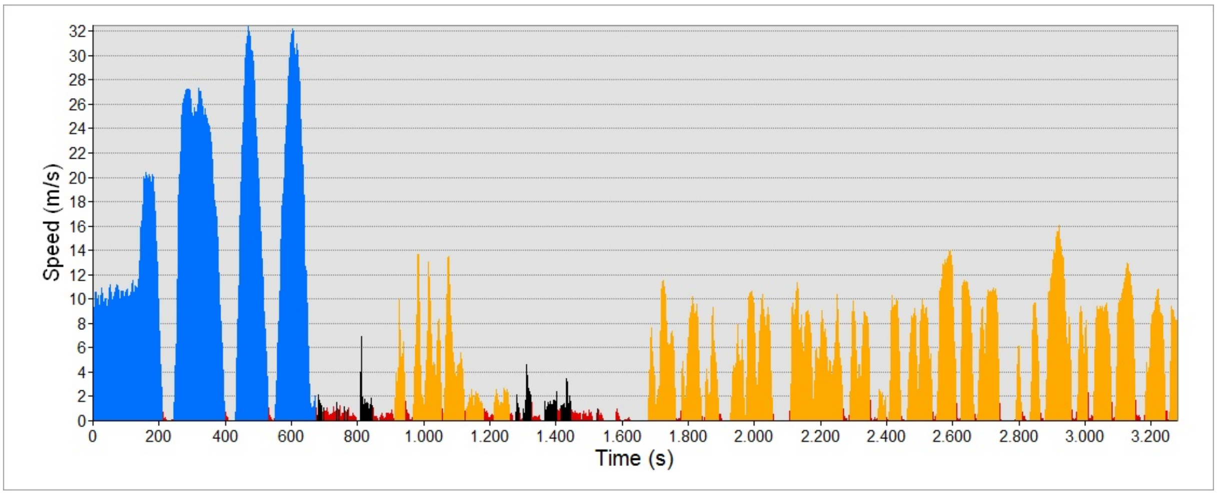


Figure 4.8: Vehicle Sub-trajectory travel modes are identified using SVMs method. Train travel mode is presented in blue, bus travel mode in yellow, walking in black and stops in red.

4.6 Discussion

This chapter presents a multi-stage method towards the automatic detection and classification of six travel modes from GPS-tracjectories. A GPS-trajectory is segmented to a sequence of movement segments by identifying stops, which on one hand make it possible to construct sub-trajectories corresponding to individual travel modes, and on the other hand charactrize the pattern of different travel modes. Thus, new parameters can be extracted based on these movement segments, such as travel time, stop rate, average travelling speed, etc. These advanced parameters enable SVMs supervised learning method to automatically classify sub-trajectories with high statistical certainty.

In the first classification stage, a number of user defined thresholds are needed for fuzzy-logic classifiers. By experiments, a setting that gives satisfactory results is applied. It should be noticed that the combination of several parameters are used, which enables the setting of wide enough ranges for each parameter and make the fuzzy-logic system not sensitive to the setting.

The presented method extracts the characteristics of different travelling behavior using speed, heading and movement pattern related parameters. It is expected that the trajectories have a high sampling rate and the degree of noise in GPS data should be not too high. Otherwise the calculated parameters will be inaccurate and cannot deliver sufficient characteristics of the behavior. In the second stage, the attributes for SVMs should be derived from at lest 2 movements, thus short trajectories with only one movement are not able to be classified.

Chapter 5

Integration of GPS-Trajectories and an Existing Road Map

We consider the GPS trajectory as a measurement which represents a “digitization” of the true road. Thus the existing road data can be incrementally improved with incoming GPS-trajectories. Although the accuracy of trajectories is not too high, due to the high number of measurements an improvement of the quality of the road information can be achieved.

In this chapter, the method of integrating GPS trajectories and an existing OSM road map towards a more accurate, up-to-date and detailed road map is presented. The integration of GPS-trajectories mainly has to deal with the high degree of noise resulting from the low quality of the GPS measurements. This makes it on the one hand difficult to discern and separate nearby roads and on the other hand also to reconstruct the underlying structure in the road geometry, e.g. the number of lanes. Besides geometrical matching, integration of travel modes and statistical classification method are also applied in order to achieve a more precise result. Thus, the assumption is that there is an existing road element, which can be geometrically imprecise. Also, the structure of the roads might be unknown, e.g. in terms of number of lanes.

The process can be summarized as following (Figure 5.1): First we profile the existing road by a sequence of perpendicular lines and get the road’s candidate sampling trajectories which intersect with the profile. Then we identify the corresponding sampling trajectories of the road using travel modes, geometric and clustering matching method. Finally, the new road centerline is estimated from its corresponding trajectories. Due to errors contained in the GPS data, the challenge here is separating GPS-trajectories appropriately and matching them to their corresponding roads from which they are collected, especially if several roads are nearby. In addition to the geometry of roads we also present the method to mine attribute information from GPS trajectories, such as number of lanes, traffic constrains.

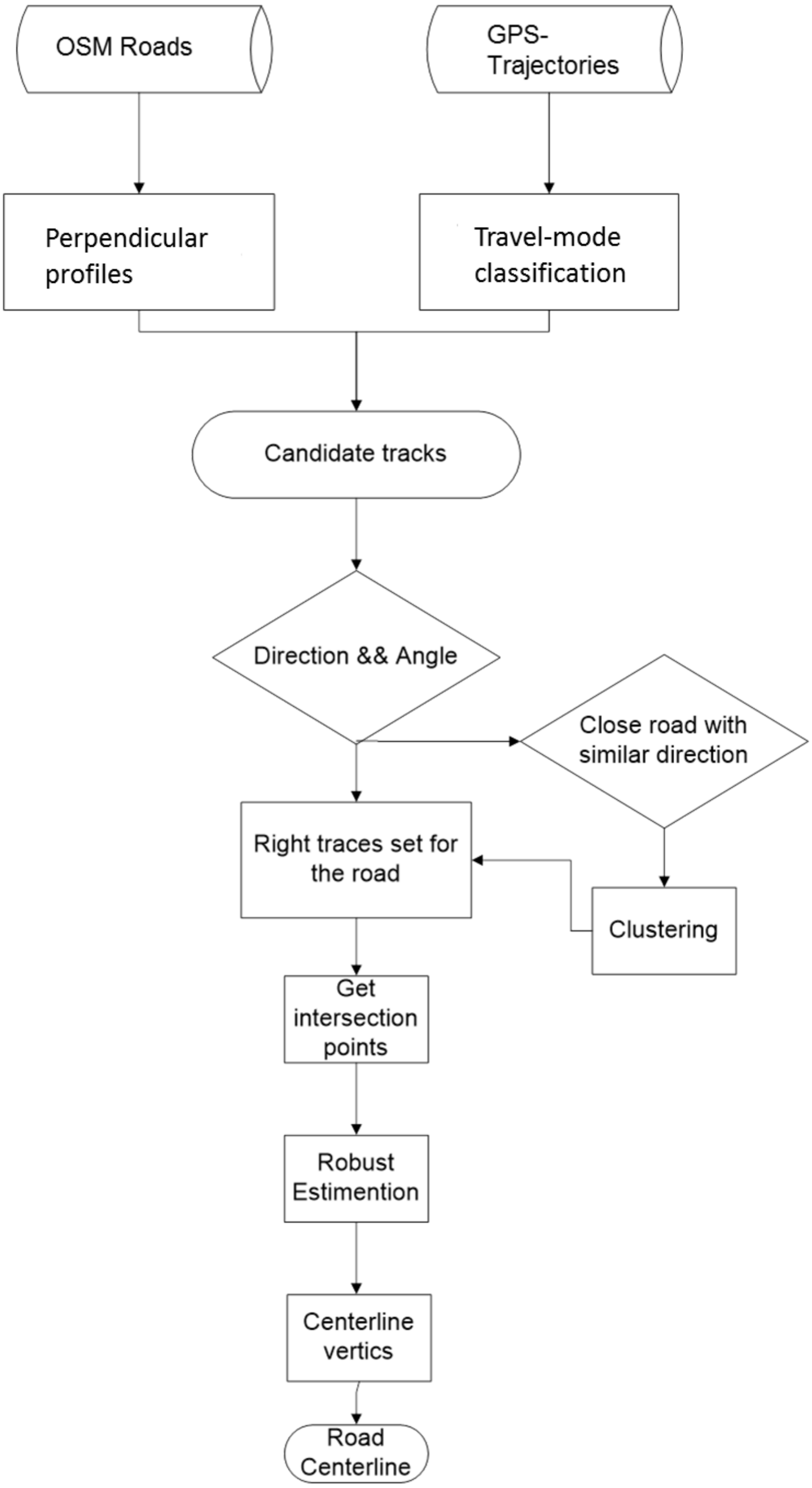


Figure 5.1: Workflow for extracting of road centerline.

5.1 Preprocessing

GPS-trajectory data are preprocessed before matching them with reference roads (OSM road map). The GPS data consist of sequential GPS points, which have latitude, longitude and sometimes a time stamp. GPS points are linked to form trajectories according to time sequence. In some cases, there are unreasonable links between different trips. Therefore, we split GPS trajectories into individual trips. We split the trace wherever the distance between two adjacent points is larger than 200 meters or the change of heading is larger than 90 degrees. A distance between two adjacent points is unlikely larger than 200 meters in one trip. The large heading changes often happen when the vehicle travels at a very low speed or it stops for traffic reasons, such like traffic jams or traffic lights. These points with very low speed have a high degree of error in position, and are not meaningful information for the integration task. Thus they are eliminated by breaking the links between them. Figure 5.2 shows the GPS-trajectory data before and after preprocessing.

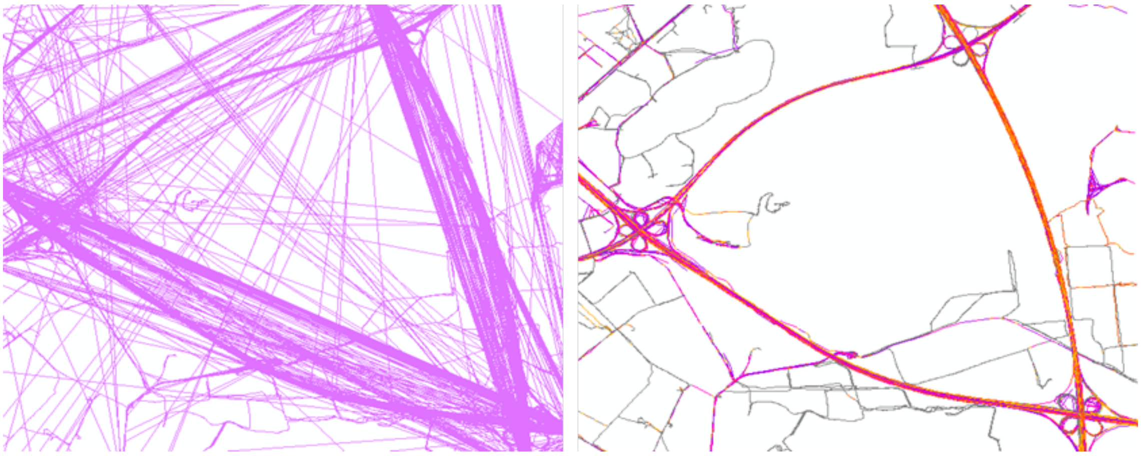


Figure 5.2: GPS trajectories, before and after preprocessing.

5.2 Extraction of road centerlines

The challenge in interpreting and integrating the GPS-trajectory data is firstly to determine the centerline from multiple representatives of GPS trajectories. Furthermore, if several roads are nearby, they have to be separated appropriately.

We consider the individual GPS-trajectories as measurements which are associated with a certain error. The “true” geometry is then derived by averaging all trajectories corresponding to one road. As depicted in Figure 5.3, in order to start the process, we use the road map from OSM as initial prior information. In order to determine the road center line, we sample it at certain distances, by putting profiles perpendicular to the reference road. The intersections of the profile with the GPS-trajectories deliver possible sampling points for the road centerline, and are then processed to match the prior road.

The matching method consists of three parts: matching according to travel modes, geometrical matching and clustering method. After the corresponding trajectories are matched to the prior road, they are used to calculate the new road centerline.

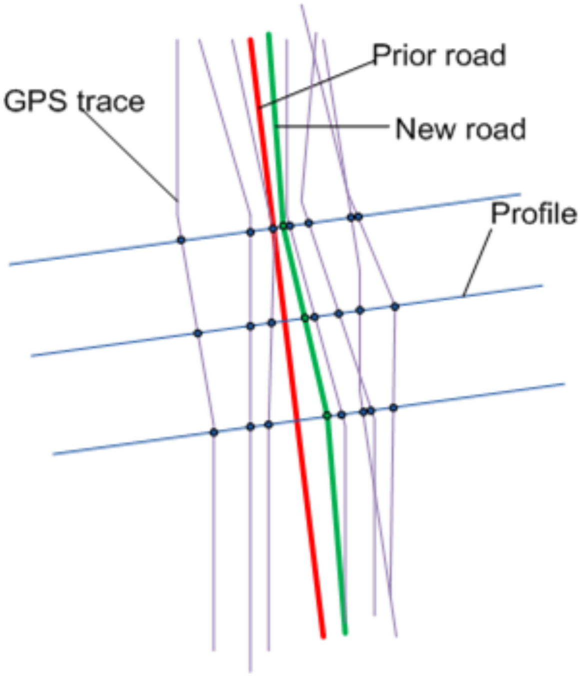
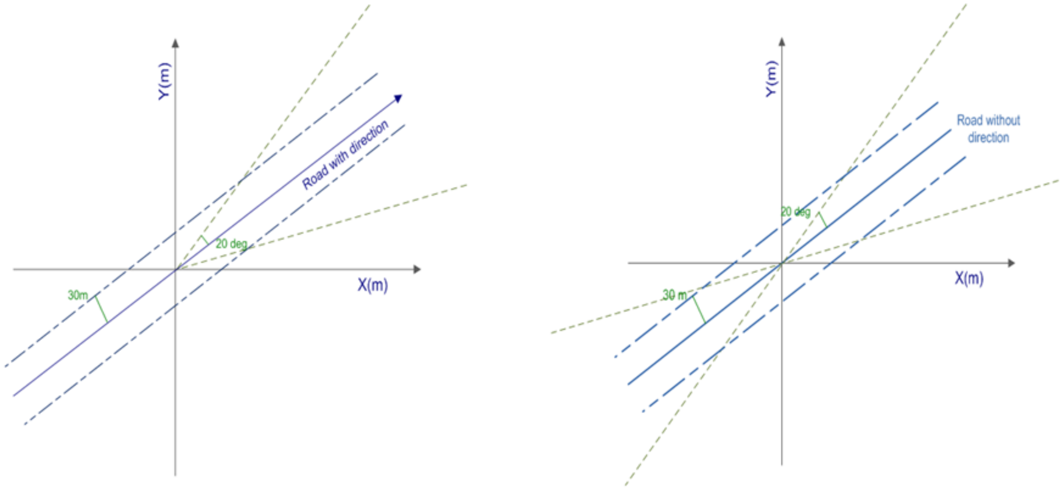


Figure 5.3: The process of extracting road centerlines.

5.2.1 Matching using geometrical method

There are three geometrical conditions we used to find corresponding trajectories of a prior road: distance to the road, direction, heading difference between the trajectory and the prior road. The conditions are shown in Figure 5.4.



The condition when the road has a direction

The condition when the road has no direction

Figure 5.4: Geometrical conditions for matching corresponding trajectories to a prior road.

The prior road map (OSM map) uses sequences of line segments that connect coordinate points which represent the centerline geometry. If a road's "ONEWAY" attribute is yes, the road has a direction that accords with the sequence of its line segments. Otherwise, the sequence of line segments does not indicate the road's direction. We then say the road has no direction restriction and it means that the vehicles can drive in both directions on it.

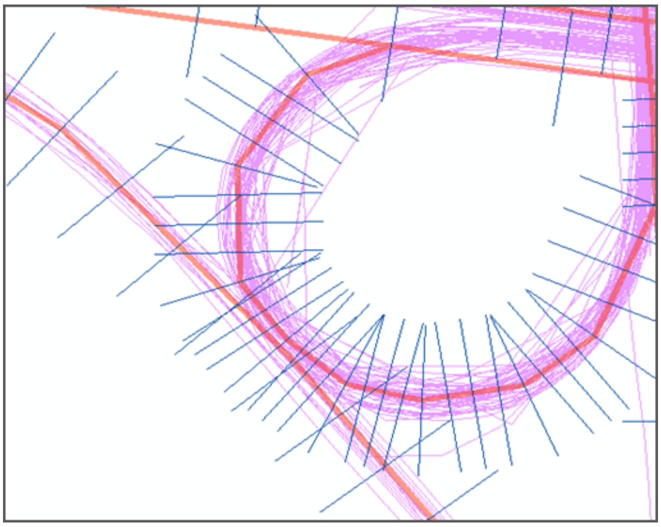


Figure 5.5: Getting candidate trajectory set for the road using profiles perpendicular (in blue) to prior road's centerline (in red).

First, as shown in Figure 5.5, we determine profiles along the road and with a width of 30 meters. We try to use wide enough profiles to tackle the errors in the prior roads and make sure that all possible trajectories for the road are included. We try with 10 meters, 20 meters, and 30 meters buffers. We find that a 30 meters buffer is suitable to select possible trajectories, considering there might be error in prior roads. The profiles are perpendicular to the road segment's direction that they belong to. The trajectories that intersect with the profile are candidate trajectories for the road.

Second, trajectories are removed from candidate trajectory set if the difference of headings between the trajectory and the road is larger than 20 degrees. Here we also make experiments to make sure that the angle threshold is neither too small to neglect right trajectories nor too large to select wrong trajectories.

At last, if the prior road is a one-way road, only the trajectories having the same direction as the prior road remain in the candidate set. Using this geometrical matching method, the trajectories can be assigned to their corresponding road if there is no other neighboring road that is nearby and has similar direction.

However, if there is a neighboring road that is close enough and has similar direction as the current target road, trajectories cannot be separated from trajectories of its neighborhood road using only the presented geometrical matching method. Figure 5.6 shows such a case where the two roads are closely located and have the same direction. The resulting roads gather together since the trajectories for them are not separated. In order to separate also such cases, in addition to the above measures we use a clustering method, which is described in the following section.

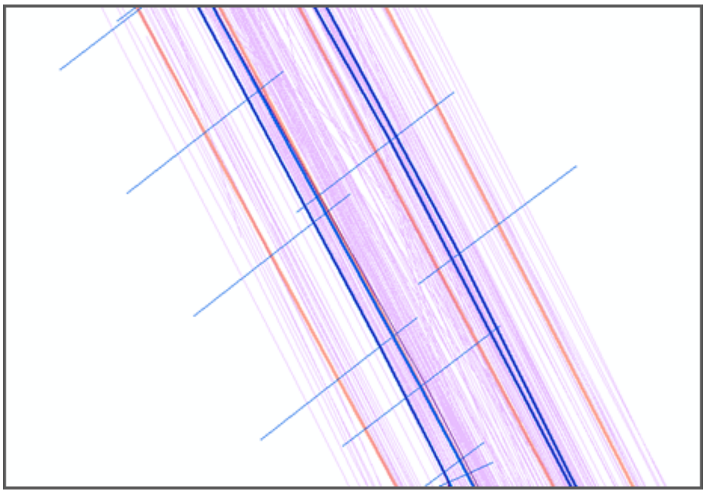


Figure 5.6: An example where the two parallel roads are closely located. The estimated centerlines (in blue) cannot be separated using geometrical matching method.

5.2.2 Matching using fuzzy c-means method

When two roads are close to each other and have similar directions, it is difficult to assign the trajectories to the right road using only geometrical methods. In this situation, we use a fuzzy c-means clustering method to separate them. The fuzzy c-means algorithm is very similar to the k-means algorithm. In fuzzy c-means clustering, instead of belonging completely to just one cluster, each point has a degree of belonging to each cluster.

The procedure of the fuzzy c-means clustering method involves an optimization of an objective function, that is:

$$c_i = \frac{\sum_{j-1}^{n} u_{ij}^{m} x_j}{\sum_{j-1}^{n} u_{ij}^{m}} \tag{5.1}$$

where c_i represents the i_{th} cluster center, u_{ij} denotes the degree of belonging of j_{th} point to i_{th} cluster center, parameter $m > 1$ is a weighting exponent that determines the amount of fuzziness of the resulting classification, $d_{ij} = \|c_i - x_j\|$ is the Euclidean distance between j_{th} point and i_{th} cluster center, where x_j is the j_{th} point.

With fuzzy c-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \tag{5.2}$$

The degree of belonging is related to the inverse of the distance to the cluster center, and the coefficients are normalized with parameter m so that their sum is 1.

$$J(U, c_1, \dots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j}^{n} u_{ij}^m d_{ij}^2 \tag{5.3}$$

The algorithm first assigns two initial random cluster centers and randomly sets initial coefficients to each point for being in the clusters. Then the algorithm computes the center for each cluster using formula 5.2 and recalculates the coefficients of being in the clusters for each point using formula 5.3 iteratively until the algorithm has converged, that is the change of the objective function 5.1 between two iterations is less than a given sensitivity threshold.

The clustering result is sensitive to the initial cluster centers. In order to get a better result, we set the point that is nearest to the start of the perpendicular line as one initial cluster center and set the point that is nearest to the end of the perpendicular line as the other cluster center. In this way, we get the maximum separation of the hypothesized two clusters.

First, we find roads that have nearby parallel roads with similar directions by checking their neighborhood. If one road has two or more close roads with parallel directions within 30 meters buffer from it, we narrow the width of profile line according to the distance between it and its nearest neighborhood road. In this way we ensure that there are only two clusters to be found. We then classify them into the following classes:

- Road has its neighborhood road on its left.
- Road has its neighborhood road on its right.

As described above, after the matching method we get a series of points that trajectories intersect with prior road's perpendicular line. If the road is one of the two types, we separate them into two clusters using fuzzy c-means algorithm. We get two cluster centers and a matrix of the degree of belonging to each cluster for each point. If the road is of type 1, trajectories belong to the cluster near the end of the perpendicular line are sampling traces for the prior road. Otherwise, trajectories belong to the cluster near the start of the perpendicular line are sampling traces for the road. Then we look into the degree of membership matrix and select traces whose degree of belonging for that cluster is larger than 0.5. In order to get a more reliable result, we may select trajectories with a higher degree of membership.

5.2.3 Construction of new centerlines

After the matching step, trajectories are assigned to their corresponding road. We get the intersection points of these trajectories with the road's profiles. Then we use a robust estimation method to select the points within 95% confidence interval, and estimate the new road center vertices, which are connected to construct the new centerline. We also add the number of trajectories that are used to construct the road and estimated standard deviations to represent confidence in the vertices.

5.3 Estimation of number of lanes

We also estimate the number of lanes for each road. GPS trajectories are expected to cluster near the center of each lane. However, as shown in Figure 5.7, due to the errors of the GPS data – and possibly also due to the lacking of enough sampling trajectories – trajectories are not clustered for each lane. This can be seen in Figure 5.8, which shows a section of a three-lane road and the corresponding number of trajectories in terms of a histogram. Thus more sophisticated approaches like proposed by CHEN and KRUMM (2010) cannot be applied. The spread of the trajectories for a road can be modeled as a Gaussian distribution. We therefore consider the mean of the Gaussian distribution as the center of the road. The standard deviation σ of the Gaussian distribution can be put into a relation to the width of the road (see Figure 5.7 and Figure 5.8). After an analysis of our data we concluded that the width roughly corresponds to 2σ.

The placement of the vehicle can be in a wider range in a multi-lane road than in an one-lane road. Therefore, the width of the road affects the spread of the traces and this can be reflected by standard deviation σ of Gaussian distribution. Through analyzing our test data set we found out that, if a road has more than 2 lanes, the width of each lane is about 3.5 meters, and the width of a normal one-lane road is about 5 meters. Thus we calculate the number of lanes using the following method: if the value of 2σ is smaller than 5.5, the road is a one-lane road. If 2σ is larger than 5.5, then the number of lanes corresponds to$2\sigma/3.5$. In the case of Figure 5.8, 3 lanes have been extracted from the distribution of the 56 sampling trajectories for the road.

$$n = \begin{cases} 1 & 2\sigma < 5.5 \\ {}^{2\sigma}\!/_{3.5} & 2\sigma \geq 5.5 \end{cases} \tag{5.4}$$

The described method of estimation of number of lanes requires a large amount of trajectories, which cluster closely around the centerline of the road. Noisy data may lead to inaccurate high number of lanes, since the standard deviation of the trajectory distribution reflecting the quality of the data may be too large.

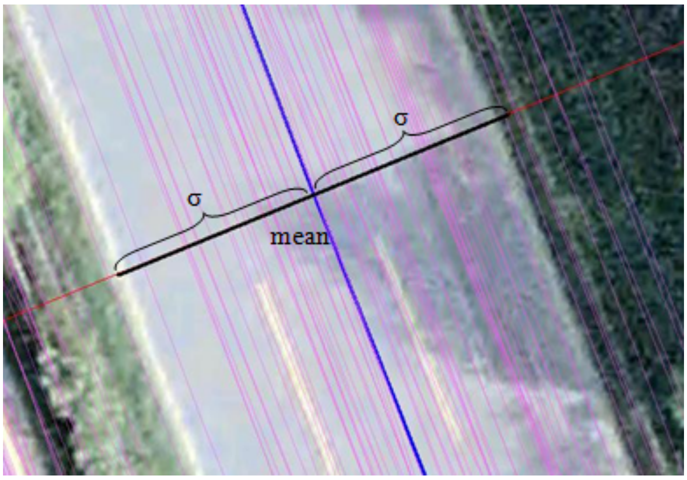


Figure 5.7: Standard deviation σ as a measure for the width of the road.

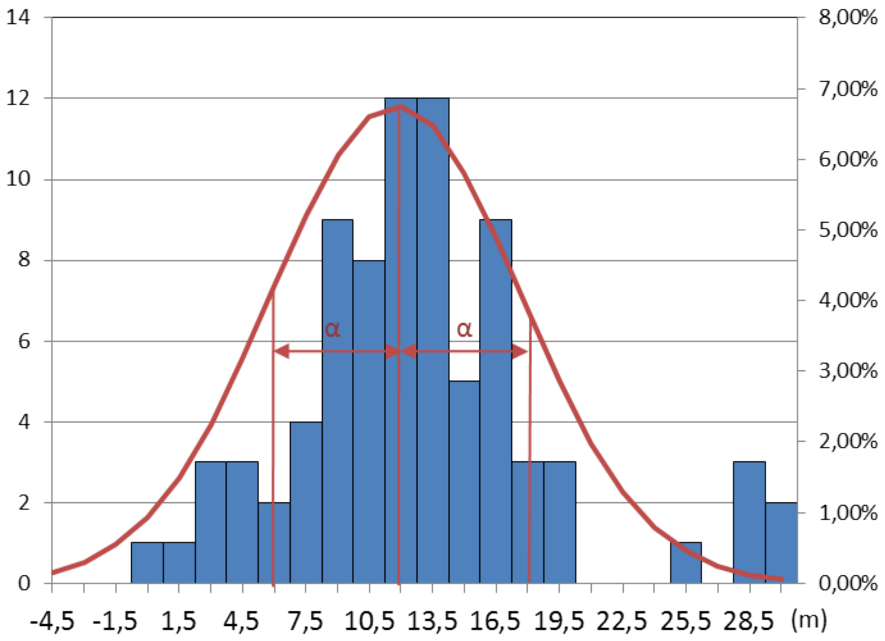


Figure 5.8: The distribution of GPS trajectories of a road can be modeled as a Gaussian distribution.

5.4 Discussion

In this chapter, we have presented a method for the improvement of existing road data with massive amounts of GPS-trajectory data, possibly of low quality. We also extract additional attribute information about the number of lanes from GPS trajectories. The method is summarized as follows:

- A geometrical matching method finds corresponding trajectories for prior road according to the distance, the direction and the angle between the trajectory and the road. By experiments, the setting of 30 meters and 15 degree thresholds for distance and angle are found to give satisfactory results.

- We use fuzzy c-means clustering method to separate trajectories for two roads that are closely located and parallel to each other. Here we restrict the clustering method to specify two clusters, since it is unlikely to have more than two parallel roads in normal conditions, especially when the type of travel mode of the trajectory is identified.

- We focus on the construction of the centerline from trajectory data. The topology of the resulting roads has not been taken into consideration. This especially influences the situation at road junctions.

- The approach can be extended to better compensate for changed or even unknown prior information, i.e. missing roads or newly constructed roads. It is possible to use one trajectory as hypothesis for the prior road to generate the new road centerline.

Chapter 6

Behavior Detection from GPS-Trajectory Data

Anomalous patterns detection refers to the problem of finding the part of the trajectory that showing the extraordinarily possibility of not conforming to expected behavior. In the following, the determination of anomalous behavior from trajectories is described. We focus on the GPS trajectories finding where the driver is encountering navigational problems, i.e., performing a detour, finding parking places, tending to lose the way, to name a few.

The chapter is structured as follows. In Section 6.1, we introduce the anomalous behavior detection from individual trajectories. The Anomalous behavior in trajectories and the features we employed to recognize it are introduced. Then the Markov model adapted to the trajectory data and the recursive belief filter are demonstrated. Collective behavior detection in groups of trajectories is introduced in Section 6.2.

6.1 Detection of anomalous behavior from GPS-trajectory data

The characteristics of anomalous behavior can be extracted via long-term features, which are considered as a sequence of control steps the driver made to choose the route approaching his/her destination. The long-term features can be extracted from the trajectory data, which include turns and their combination, degree of detour and route repetition.

We use an extended Markov chain to model the trajectory integrating these long-term features. A recursive Bayesian filter is applied to process the Markov model and deliver an optimal probability distribution of the potential anomalous driving behavior over time. The proposed filter performs unsupervised detection in single trajectories with the described features. No training process is required to characterize the anomalous behavior.

6.1.1 Anomalous behavior and features

In this thesis, anomalous patterns in trajectories refer to the patterns that do not conform to expected behaviours. In contrast to a normal drive from the start spot to the predetermined destination, anomalous behavior may happen in many various situations, e.g., taking a wrong turn, getting lost, road-block, temporary stopover, etc. the anomalous behavior and features cannot be determined by a single time stamp but a period of driving must be taken into consideration. Unlike during normal driving, when the vehicle is performing an anomalous behavior, frequently turning, detouring and coming back to a previous road may happen to a large degree. Thus, a long-term observation is performed to extract the required features: (1) turns, their combination and density, (2) the degree of detour and (3) the route repetition from the original data. These high-level as well as long-term measurements - behavior features are used as the ``observations" of the underlying states.

Turns combination and density

Turning is the most basic feature in trajectory data. The route selection of a travel can be seen as a series of turns that are determined by the driver during driving to the destination. Although a single turn is not indicating any anomalous behavior, their combination and density can deliver some anomalous patterns like forming a detour/loop and unusually intensive turns.

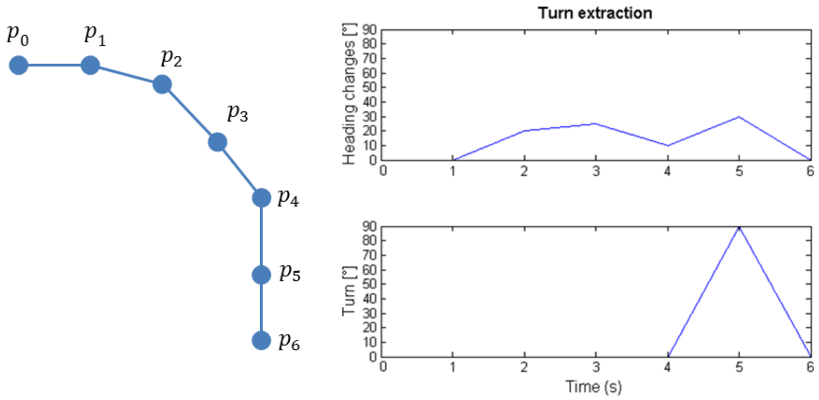


Figure 6.1: Turn extraction: the accumulated heading changes are extracted as a turn feature at the last time stamp.

A turn is normally not captured by a single GPS point, but is accomplished with several time stamps (several seconds in normal condition). We use an incremental method to extract turns, which is similar to the method used for identifying stops (cf. 4.2.2). As shown in Figure 6.1, when the heading change at a time stamp is detected as larger than 1°, it is treated as the first time stamp where a turn starts (p_1). The values of heading changes are summed until the heading change is less than 1° at a time stamp, where the turn is finished (p_5). And the turn is ``marked" with the total heading change value at the last time stamp when the turn has been finished (i.e. 90° at p_5). A turn that is measured clockwise from the previous direction is defined as positive, while in the opposite way it is seen as negative.

We use the total absolute heading change of 40° as the threshold to determine a turn. In other words, if the heading change is less than 40°, it is not marked as a turn. However these heading change values are maintained in the data and are used to determine whether the car is performing a detour or not.

Figure 6.2 shows two special situations of extracting turn features. The left panel of Figure 6.2 depicted the error in the data shown as a jump in position. Although large heading changes are detected, the direction of the trajectory after the jump is not changed. Thus no significant heading change is accumulated for these time stamps and no turn is marked. The right panel of Figure 6.2 shows a circular road, as the heading changes continuously the turn is not finished until the circle is passed and heading change of one time stamp is less than 1°. Although the heading change after the circle is about 270°, it is treated as it simply turns left and the value -90° is given for the turn.

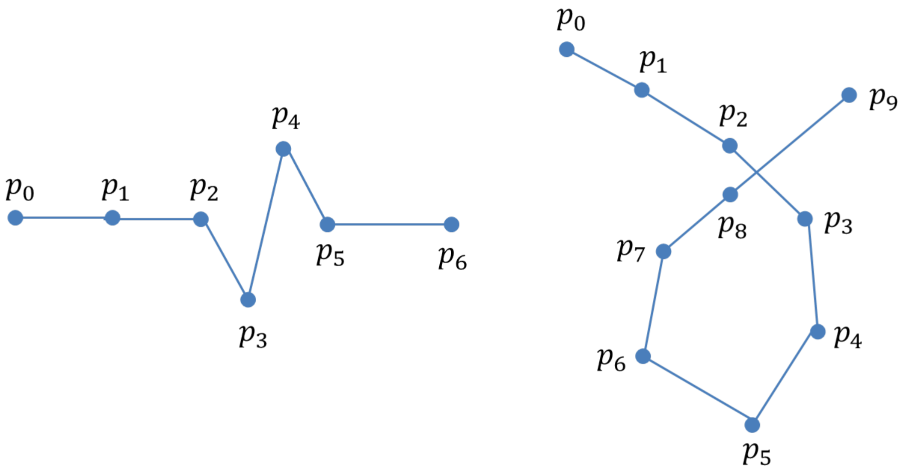


Figure 6.2: Two special cases showing the turn extraction. The left part of the figure shows the error in the data and the heading changes are ignored. The right panel shows a circular road situation where heading changes continuously and a left turn is extracted.

In addition to the detection of a single turn, a perspective with an even longer term is taken to observe and evaluate the combination and density of multiple turns. Intensive sequential same direction turns, e.g., double or triple left turns, have more impact on the belief of anomalous behavior than the sequential different turns because they are implying a potential detour or the tendency of looping.

Detour factor

When the driver meets traffic issues, e.g., road-blocking and traffic jam, or fails to find the correct or best way to the destination, a detour often can be expected. We use the detour factor to quantify the degree of detour as an important anomalous feature.

A detour is detected when the trajectory tends to go backwards according to a start point, or in other words, heading change from a start point is larger than a predefined threshold. Here we define $135°$ as the threshold (other than $180°$) to include more detours. The detour factor is then calculated to quantitatively evaluate the degree of the detour. Turns are the basis for determining a detour in the trajectory. A detour starts with a turn and ends up with another different direction turn.

Figure 6.3 gives a simplified example how we define a detour and calculate the detour factor. From a start point, if the trajectory tends to go backwards, or in other words, the heading change is larger than 135° , it will be treated as a detour and the detour factor will be calculated for all the time stamps in the backward segment.

As shown in Figure 6.3, the time stamps that are marked as turns (p_3, p_6, p_9, p_{12}) are start points for detour checking. And heading change values from the start point to the current turn are accumulated, and if the heading change from the start point to the current turn is larger than 135° a detour is identified. Let p_3 be the start point, turn values of p_6 (90°), p_9 (90°) are accumulated to 180°, and when checking p_{12} (−90°) the total heading change is then decrease to 90° (less than 135°) and the detour is finished at p_{12}. The points of the trajectory $\mathrm{T}[p_3, p_{12}]$ are then assigned the detour factor as depicted in the bottom panel of Figure 6.3.

The detour factor of an individual point is calculated as the ratio of the length of the trajectory from the start point (solid red) and the direct distance between the start point and the current position (green dashed line). The detour factor values are given to the segment until the heading change value decreases to less than 135°. A value of 1 is then given to the rest points meaning no detour.

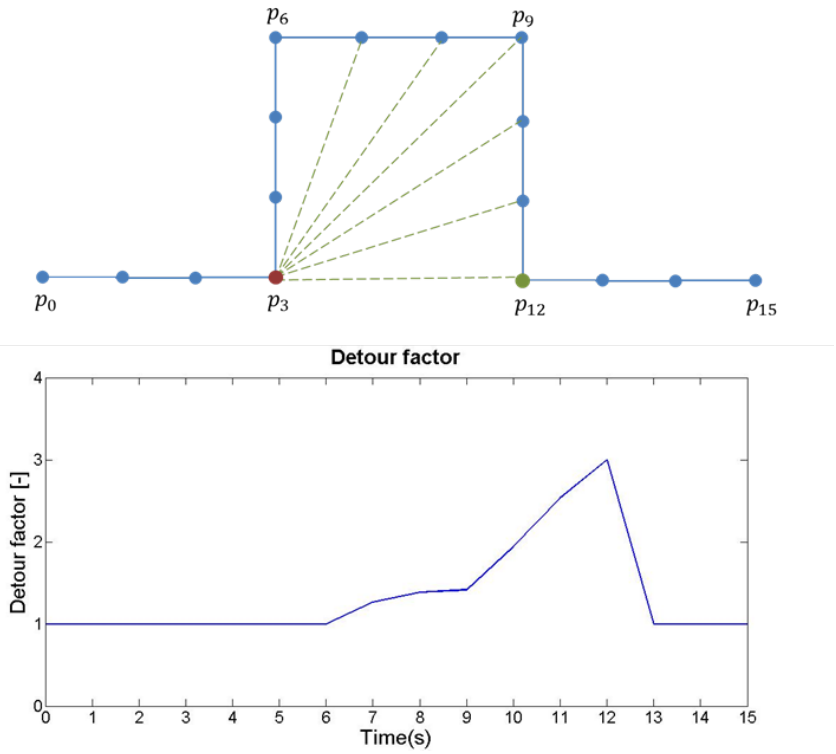


Figure 6.3: Detour detection and detour factors calculation. A detour is detected when the heading changes accumulate to more than 135°. Start and end points of the detours are represented as red and green circles respectively. Detour factors are calculated by the ratio of the length of the trajectory and the direct distance between the current point and the start point (green dashed line).

A small detour sometimes may be a part of a large detour. If this situation occurs, the detour factor for each point is then calculated for each detour respectively and the largest detour factor value is assigned to the point. As shown in Figure 6.4, segment $\mathrm{T}[p_3, p_{16}]$ is a big detour and the segment $\mathrm{T}[p_8, p_{16}]$ inside $\mathrm{T}[p_3, p_{16}]$ is also considered as a detour. Detour factors for segment $\mathrm{T}[p_8, p_{16}]$ are calculated separately respects to start point p_3 and p_8, and a larger value is kept as the detour factor for the points of segment $\mathrm{T}[p_8, p_{16}]$.

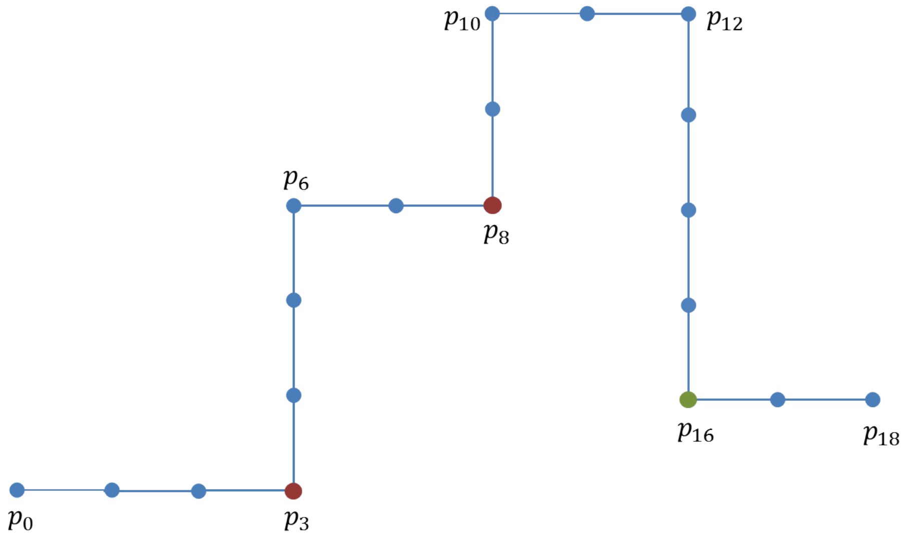


Figure 6.4: A complicated detour example: a small detour $\mathrm{T}[\mathrm{p}_8, \mathrm{p}_{16}]$ is a part of a large detour $\mathrm{T}[\mathrm{p}_3, \mathrm{p}_{16}]$. The detour factors for the small detour are calculated twice according to the two start points. Start and end points of the detours are represented as red and green circles respectively.

Route repetition

The most prominent feature in an one-way trajectory is the route repetition, i.e., the vehicle goes back to the same road part, from either the same or opposite direction, on its way to the destination. Route repetition with the opposite direction is mostly the result of performing an U-turn while that with the same direction often happens after driving a loop. The current trajectory segment, i.e., between the current and the last steps, is repeating a former route when any prior trajectory segment(s) fall inside the buffer of the current segment and approximately parallel to it.

6.1.2 The extended Markov chain

A Markov chain is commonly used in the modelling of a chain of linked events over time sequence. The Markov chain is the basis of Bayes filter variants, e.g., Kalman filter, and can be easily considered as an appropriate model for trajectory data.

The belief of anomaly is not observable or measurable and it is modelled as a discrete hidden sequence of state X_k using the Markov model. The extracted long-term features of turns, detour factor and repetition can thus be considered as the measurements/observations Z_k in Morkov model. It is assumed that each observation Z_k is depends on a discrete hidden state X_k and the sequence of hidden sequence state is distributed according to a Markov process, as shown in Figure 6.5.

Let X be the unobserved state (here the probability of anomaly) and Z the measurements (turns, detour factor and repetition), the Morkov chain as the process model is presented in Figure 6.5.

The Markov chain $X = \{x_1, \dots, x_n\}$ follows the Markov assumption, i.e., the probability of the current state given a limited number of previous ones is conditionally independent of the other earlier states:

$$p(x_k | x_{k-1}, x_{k-2}, \dots, x_{k-m}, \dots, x_0) = p(x_k | x_{k-1}, x_{k-2}, \dots, x_{k-m}) \quad (6.1)$$

with $\mathrm{m} < k$. The measurement $Z = \{z_1, \dots, z_n\}$ at each state is dependent not only on the corresponding state, but also several previous states:

$$p(z_k | x_k, x_{k-1}, x_{k-2}, \dots, x_{k-m}, \dots, x_0) = p(z_k | x_k, x_{k-1}, x_{k-2}, \dots, x_{k-m+1}) \tag{6.2}$$

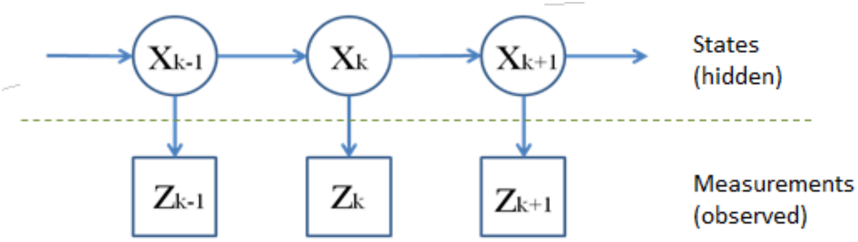


Figure 6.5: The Markov chain integrating long-term features.

6.1.3 The recursive belief filter

We detect anomalous behavior from trajectory data using a variant of recursive Bayesian filter. The pattern detection is applied to individual trajectories and find where the driver is performing anomaly that tends to take a detour or repeats previous route. No previous learning process is needed to distinguish normal and anomalous behavior.

The proposed filter is a simple variant of recursive Bayesian estimator keeping the dynamic property and the prediction/updating scheme. Although normally the prediction and updating steps work alternately and provide the required inputs for each other, either of them has also the probability to be skipped. In this work both of these cases will happen:

- We are using 3-dimensional measurements (behavior features) for the updating. We assume the features are independent of each other. Sometimes no feature is derived at one time stamp when no turn, detour and repetition occur. More than one feature is extracted at individual time stamps may also occur. With the assumption that the features are independent to each other the updating step is performed multiple times if there are more than one measurement before next prediction.

- These long-term features, however, cannot be constantly observed. In the interval of the given observations, the prediction will be performed solely for multiple times.

Prediction

Let us represent the state (belief of anomaly) at time k by random variables x_k. at each time stamp k, the uncertainty is represented by a probability distribution over x_k called belief $Bel(x_k)$. The key idea of Bayes filters is to sequentially estimate such beliefs over the state space conditioned on the long-term features. The sequence of time indexed long-term feature observations are denoted as $z_{1:t}$. The belief $Bel(x_k)$ is then defined by the posterior density over the random variable x_k conditioned on all sensor data available at time t:

$$Bel(x_k) = p(x_k|z_{1:k}) \tag{6.3}$$

The prediction step calculates the total probability, i.e., the integral of the products of the transition probability $p(x_k|x_{k-1})$ and the probability of the previous state $Bel(x_{k-1})$ over all possible x_{k-1}. The integrating over x_{k-1}follows Chapman-Kolmogorov equation:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})Bel(x_{k-1})dx_{k-1} \tag{6.4}$$

In this case, we have only one variable, i.e., the belief of anomaly, to be estimated and in principle it cannot be observed based on any current measurements. We assume that the anomalous behaviors are rather transitory than the normal drive and, therefore, use a simple exponential decay to predict the belief of the next state:

$$x_k = x_{k|k-1} = F.x_{k-1} + w_k \tag{6.5}$$

where $F = e^{-s.k}$; $w_k \sim N(0, \sigma^2)$.

F simulates the decay given to the belief of anomaly along with the driving. A Gaussian noise is added by w_k. k is used to count the number of the previous state(s) without new anomalous feature(s) being reported. The accumulation of k enables the belief decays rapidly after the driver performs normally. The decay tendency can actually be tuned by the factor s. Generically we give no weight to k for each step, i.e., $k = 1$, in the urban area.

The predicted state estimate is taken as prior estimate for the current state.

Updating

After the prediction now we have the prior distribution from the Chapman-Kolmogorov equation $p(x_k|z_{1:k-1})$. The posterior estimate is computed from prior distribution together with the observation on the current state using Bayes rule.

$$p(x_k|z_{1:k}) = \frac{1}{Z_k} p(y_k|x_k)p(x_k|z_{1:k-1}) \tag{6.6}$$

where The normalization constant $Z_k = p(y_k|y_{1:k-1})$ is given as

$$Z_k = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k \tag{6.7}$$

with multiple measurements.

Belief inference

We employ two simple typical cases: detour and wrong turn with simulated data to present the inference process using the proposed belief filter. Besides the information of turns, these two cases have their particular features that another one does not have. I.e., the detour case has only detour factor and no repeated route while the wrong turn case has the latter only. In this way, the influence of the individual features can be well demonstrated. Figure 6.6 presents a simple simulated trajectory with detour (left) and the extracted high-level features plotted over time (right). The bold red line shows the inferred belief of anomalous patterns over time. The possibility values are also presented in the trajectory with different colors. Please note that a green circle and a red asterisk are used to mark the start and end positions of the trajectory, respectively. The value/color of each line segment is determined by its start point. We use these two examples to demonstrate some typical situations in the inference process.

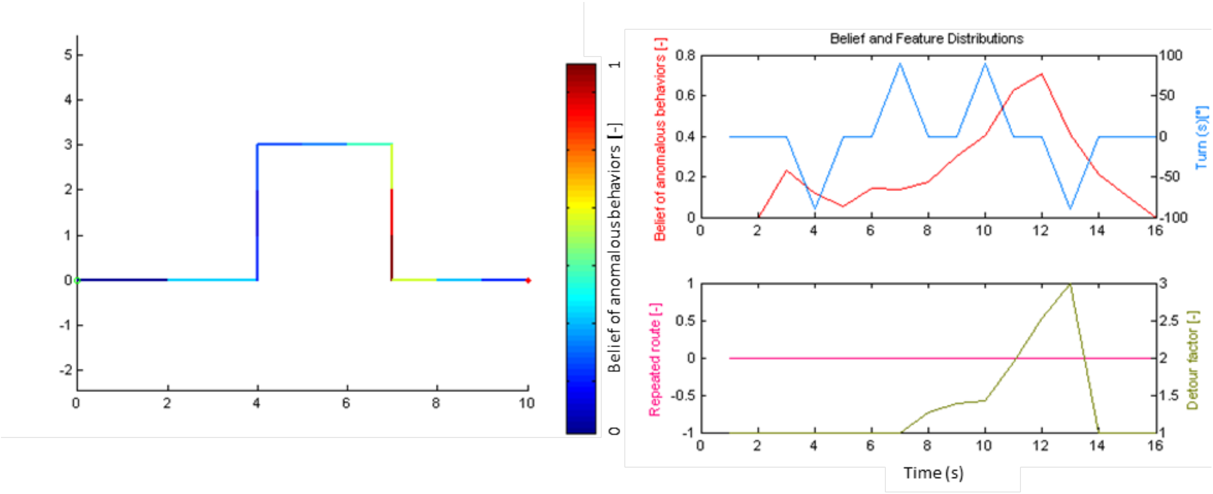


Figure 6.6: Simulated trajectory of detour (left) with start position (green circle), end position (red asterisk) and the belief of anomalous behavior shown with scaled color. Three high-level features: Turns (blue), repeated route (magenta) and detour factor (green) are plotted together with the belief of an anomaly over time (right).

- Combination of double different turns is considered normal. If the current turn has the different direction to the previous one, less probability is given to guarantee the continuous decay of the belief of an anomaly.

- Double same turns, in contrast, mean potential detour or even looping. Probability gain is added when the second turn happens.

- Detour factor increases and reach the maximum value when the detour is finished. The belief of anomaly has the peak value at this time as well.

- After the detour, the belief of an anomaly has a fast decay.

Another typical case of a road repetition is shown in Figure 6.7. The repeated part of the route is given the same repetition feature value of 1, as the vehicle comes back to the road in a different direction. The belief increases fast when the vehicle goes back to the previous road and reaches the peak value at the position where the wrong turn started. The belief decays to a low value when the vehicle gets back to the original road.

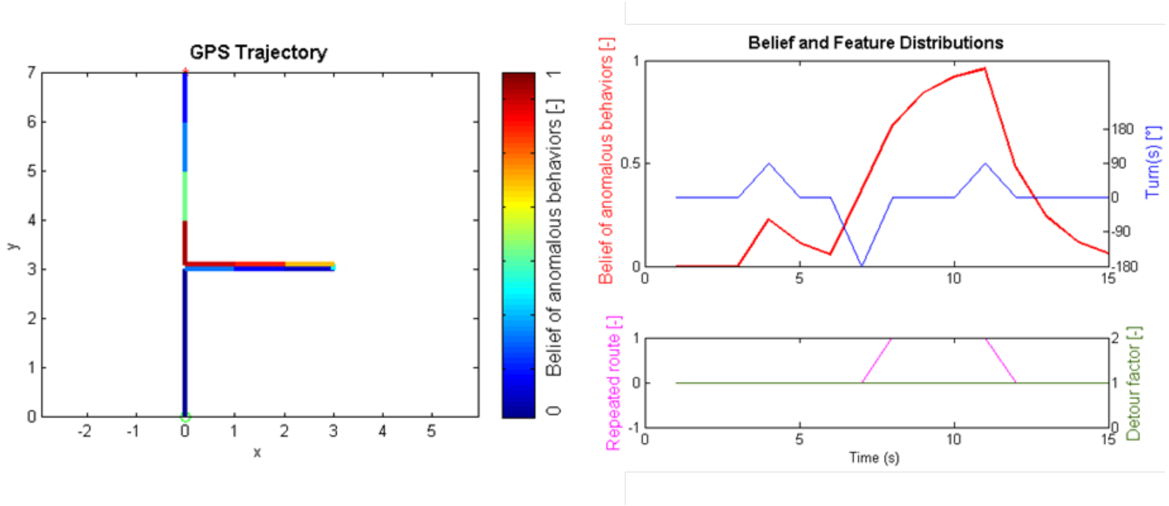


Figure 6.7: Simulated trajectory of repetition: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right).

6.2 Detection of collective behavior patterns

The described travel mode behavior classification in Chapter 4 and anomalous behavior detection in this chapter can both be categorized to the problem of behavior detection in single trajectories. Patterns in groups of trajectories can present shared beliefs and common goals with the individual personalities becoming less important. The collective behavior may reveal new characteristics beyond individual behaviour. In the following, two examples for collective behavior are described: the collective travel mode change behavior and collective anomalous behavior are investigated.

6.2.1 Collective travel mode change behavior

In Chapter 4, the method of segmentation and classification of sub-trajectories of an individual travel mode in single trajectories is presented. Changing the travel mode is an important navigational behavior of commuters. Moreover, the behavior is often repeated at specific locations frequently. The collective travel mode change behavior recognized in a group of trajectories is useful for many tasks, such as route planning, understanding mobility, etc. This is important since the collective behavior in a few trajectories is usually not sufficient to identify the pattern, and a large number of travel mode changes in a spatial region are needed. Thus, the goal is to identify clusters of travel mode changes of many subjects in terms of both space and time. In this way, first of all, locations such as park and ride stations can be identified. Another interesting application is described in the following.

One potential application is to improve route planning by making available of the following information: parking location associated with a specific address. Navigating and travelling between destinations with the help of Navigation Systems is a very common task carried out by millions of commuters daily. Route planning is mostly based on the addresses given by the traveller, and thus the creation of path and directions needed to be taken. This means that the traveller is guided directly to the address, most often the entrance of the building. Today's navigation data sets rarely contain information about parking lots, related to building entrances. This is especially relevant for large building complexes (hospitals, industrial buildings, city halls, universities). A fine-tuned route tailored for the driver requirement, e.g., parking the car close-by to destination, is required in such cases to save time and prevent frustration.

All trajectories composed of car accompanied by walking travel modes are identified, so the extraction of parking locations that are associated with specific buildings can take place. Such information can later be added to the road navigation maps used by the route planning scheme to enable the construction of a more fine-tuned optimal and reliable route that will prevent subsequent detours for finding parking places. The conceptual approach for the detection of this pattern is described in the following.

To automatically detect the parking places, frequently visited travel mode change locations should be found. First, change points are identified where travel modes are altered from car to walking - or vice versa. Instead of clustering these detected travel mode change points, the parking locations could be extracted using an incremental method similar to the technique for finding “interesting places” proposed by SESTER et al. (2012). If the newly found travel mode change point is contained by a potential parking place, it is included in that parking place and is used to adjust its size. If a user defined minimum count of visits is reached for a potential place, it is then identified as a parking place.

Inspired by the method proposed by CAO et al. (2005), the sub-trajectories of walk travel mode can be simplified and represented by line segments with reduced number of points. These line segments can later be used as seeds for clustering method, which cluster nearby walking trajectories according to their closeness in space to find the frequently repeated paths. Frequently repeated paths can later be associated with parking lots of one specific building.

Since the positional accuracy of the GPS is at least several meters (and worse in build-up areas), and GPS multipath signal errors and signal-loss is common in the vicinity and inside of buildings, identifying correctly the building in which the walking started/ended in is important. This stage is achieved by performing a buffering process around building features in the close vicinity of the start and end points locations identified earlier. Since parking lots are mostly situated in the vicinity of large building complexes (hospitals, industrial buildings, and such), the assumption is that if several building buffers contain the same walking positions, all can be attributed to the same parking lot position since it probably serves all these buildings. Buffer size used is derived from the positional accuracy of the building features (existing in the given database), but also from the positional certainty of the GPS signals acquired during the walking trace. Since build-up areas are of poor positional certainty, this value should be considered based on some knowledge of the errors at hand. In the following example, buffer size used had the magnitude of 10 meters, which was found to be sufficient.

An example is depicted in Figure 6.8, which shows a building buffer polygon containing traces of walking travel modes (in red) emerging from car travel modes (in blue). It is visible that the 10 meters building buffer covers in its extent all walking traces leading to/from it. Since this building has two entrances, some walking trajectories lead to its north entrance associated with the east parking lot, while the other lead to its south entrance associated with the south parking lot. Consequently, both parking lot positions are associated with this building. Also, it is visible that other buildings in the vicinity will also be associated to one (or both) parking lots extracted here.

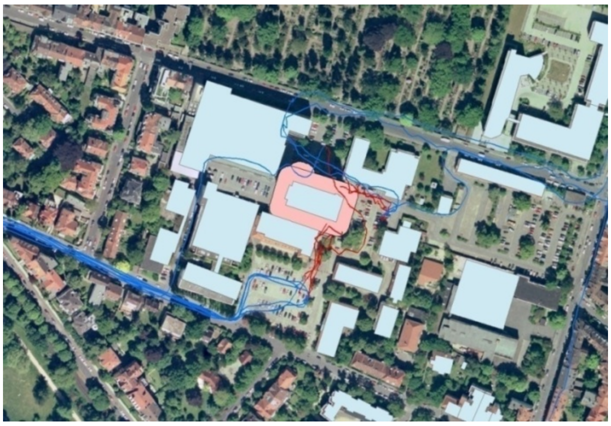


Figure 6.8: Building buffer containing walking travel modes to/from car parking place: car trajectories are represented in blue, walk trajectories are represented in red.

6.2.2 Collective anomalous behavior

The anomalous driving behavior detection presented in section 6.1 is detected from single trajectories and indicates that the driver may encounter navigational troubles. It can be expected that the anomalous behavior can be detected in a group of the trajectories that concentrate in a certain area by some traffic reasons, e.g., road-blocking or traffic accidents. The collective anomalous behavior in a group of trajectories can help us understand what happened or is happening regarding traffic conditions.

The automatic detection of collective anomalous behavior has to mainly deal with the spatial-temporal scale problem, since they are valid within various ranges of spatial and temporal scales. SOLEYMANI et al. (2014) propose a cross-scale approach, which partitions the movement space into several spatial zonings and processes the trajectory data according to a series of temporal widows. The reliable scales could be determined for different demands for both spatial and temporal domains. This method sets the direction for the future work regarding how to automatically identify collective anomalous behavior.

The underlying spatial extent could be subdivided into fine zones. When an anomalous trajectory segment is detected in one zone, this zone can be used as a seed, which can later aggregate more nearby zones containing anomalous trajectories. In this way, the spatial extent for collective behavior is expanded when necessary. The detection of collective anomalous behavior can thus be derived using an incremental method, which counts the amount of anomalous trajectories in the spatial domain. A temporal window can be used to control the incoming trajectories. It may vary to a large degree according to different application domains. For example, for traffic jam detection, a few hours may be suitable; while for road restriction detection an observation for several days is needed.

6.3 Discussion

Anomalous driving behavior is characterized by three features: turn, repetition and detour factor. A recursive belief filter is conducted for the dynamic unsupervised detection of anomalous patterns.

The anomalous detection method focuses on finding patterns in driving trajectories where the driver is likely meeting navigation problem and an assistant is needed. However, the filter could be adapted to trajectories of other travel modes by tuning the decay speed. For example, for pedestrian trajectory the decay speed may be enhanced to avoid continuously accumulation of the belief.

The detour factor feature is calculated according to the whole history of the trajectory. The method could be strengthened to better adapt to real time applications by introducing appropriate scaling scheme that abstract recent portion of trajectories. The method presented by PATROUMPAS (2003) may be a solution to query the recent motion path at different level of details in real-time.

Two examples for collective behavior detection have been presented. These examples should demonstrate the great potential of exploiting trajectories for such purposes. These applications have not yet been implemented in terms of software code. However, a possible implementation scheme has been sketched in Section 6.2. It is based on a combination of clustering methods and spatial analysis procedures, which have to be adapted and integrated.

Chapter 7

Experiments

7.1 Datasets

We use three datasets to perform experiments and test the reliability and transferability of the presented methods. The first dataset is collected by the colleagues of the Institute of Cartography and Geoinformatics while traveling in Hannover, Germany, using hand-held smartphones. A second dataset is from OSM project that is contributed by its registered users. Another open dataset is Chicago trajectory dataset that is provided by AHMED et al. (2014) to test benchmarking methods of map construction algorithms. In the following part of the section, these datasets are outlined in details.

7.1.1 Tracer trajectories collected using Smartphones

For statistical appreciation of the proposed travel mode classification methodology, a training data with supplementary information is required. For this, an Android application was programmed, which collects GPS data and reference added-data (tagging) that basically store the travel mode specified by the user. The application (in the mobile domain usually referred to as App), named Tracer, was specifically designed to be used for Android-based Smartphones. The Graphical User Interface, depicted in Figure 7.1, presents specific and easy-to-use functions. These functions include: a toggle button for starting and stopping data acquisition (left); and, a button enabling the user to select (and modify) his current travel mode (right). The user can choose from six different travel modes that are employed in this research: Walk, Bike, Car, Bus, Train, and Tram. Additionally, there exist a checkbox labeled “silent” (left), which allow the user to choose whether to be notified with some predefined events – detailed later.

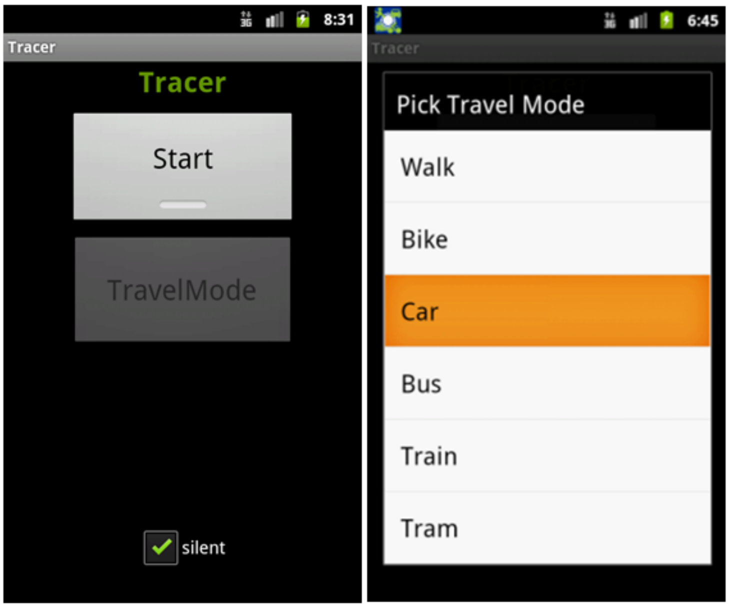


Figure 7.1: Graphical User Interface of the Tracer App: main view (left); travel mode selection (right).

Since the data acquisition is supposed to be a passive procedure, the Tracer App provides with a notification system that requires the user to focus attention on specific predefined events. The notification system utilizes all modes of user notifications provided by modern smartphones, e.g. visual, sound and haptical. Their common goal is to obtain the user's correct current travel mode. The Tracer App implements the following events:

- The constant travel mode update event forces the user to update current travel mode every 10 minutes in order to prevent from forgetting to do so.

- The GPS-signal loss event is triggered only after gaining back of signal, which was lost for more than 20 seconds. This includes cases where travel mode changes might happen without having a GPS-signal.

- Speed inconsistency cover events of derived travelling speed exceeding predefined speed limits for walking and cycling that are over 10 seconds. Thresholds used are coarse, and as such are only a type of warning.

The data collection period simulates the natural way of how people travel in their everyday life without applying any special concerns or restrictions. Total of 192 GPS-trajectories were collected in the study-area of Hannover City. 74 trajectories were contributed by commuters using Smartphones via Tracer application. And 118 car trajectories are collected using in-car GPS devices. Figure 3.2 displays the trajectories.

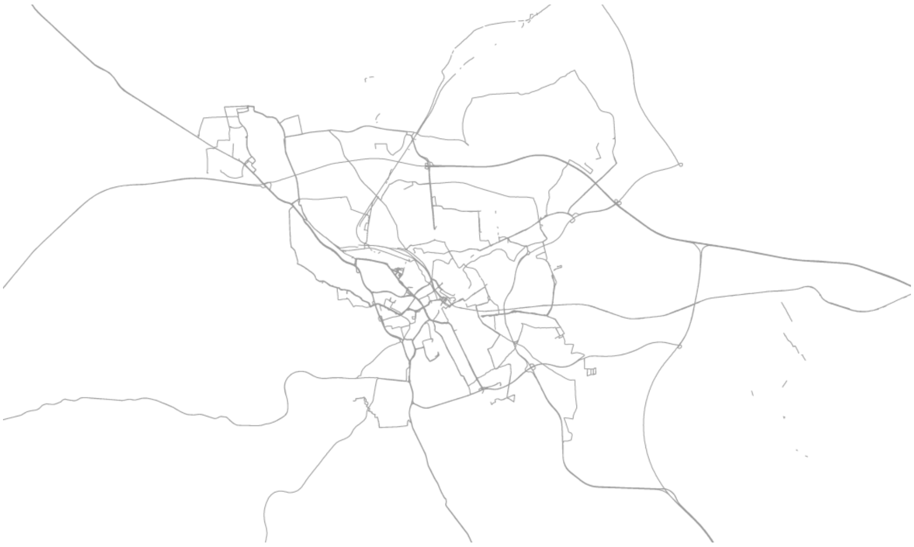


Figure 7.2: The Tracer trajectory dataset: 192 GPS-trajectories were collected in the study-area of Hannover City.

7.1.2 OSM trajectory dataset

GPS-trajectory data can be downloaded from the OSM website. The GPS data are collected by OSM users while doing their own business. The source of the data can be from varies traffic means, such as cars, trains and bicycles.

The GPS-trajectories are not distributed equally among the roads. Some roads have more corresponding trajectories than others. We observed that a typical highway has 30 to 80 corresponding traces, whereas a busy city road has less than 20 trajectories and a road in a local neighborhood has none or only a few. Even when roads arise out of the same type and close to each other, the number of their corresponding trajectories may vary noticeably.

For the major part of the trajectory data, the time stamps are not recorded. The time interval between two adjacent GPS points may vary significantly, from one second to dozens seconds. Thus the speed and heading related parameters are not able to be precisely calculated, which are key features for travel mode classification. The fact of lacking time stamps makes it impossible to correctly classify travel modes of trajectories. Thus, we use this dataset only for map refinement.

An area near Hannover city that mainly contains highways is chosen to perform the presented algorithm of road map construction. Since these trajectories are mainly derived from a highway area (Figure 5.2), where roads are not as concentrated as urban areas, we skip the travel mode matching step and trajectories are matched with their corresponding roads using only geometrical and clustering methods.

7.1.3 Chicago trajectory dataset

We also use the Chicago trajectory dataset (Figure 6.4) to test our algorithms for travel modes classification and map generation. The Chicago dataset is one of four open trajectory dataset that AHMED et al. (2014) use to test benchmarking method of map construction algorithms. The other three datasets have sampling rates larger than 30 seconds. Thus the speed and heading related parameters cannot be precisely calculated for travel modes classification using the presented method.

The Chicago dataset consists of 889 trajectories, which are obtained from university shuttles covering an area of 7km x 4.5km. The trajectories range from 100 to 363 position samples, with a sampling rate of 1s to 29s (average: 3.61s and standard deviation: 3.67s) and an average speed of 33.14 km/h.

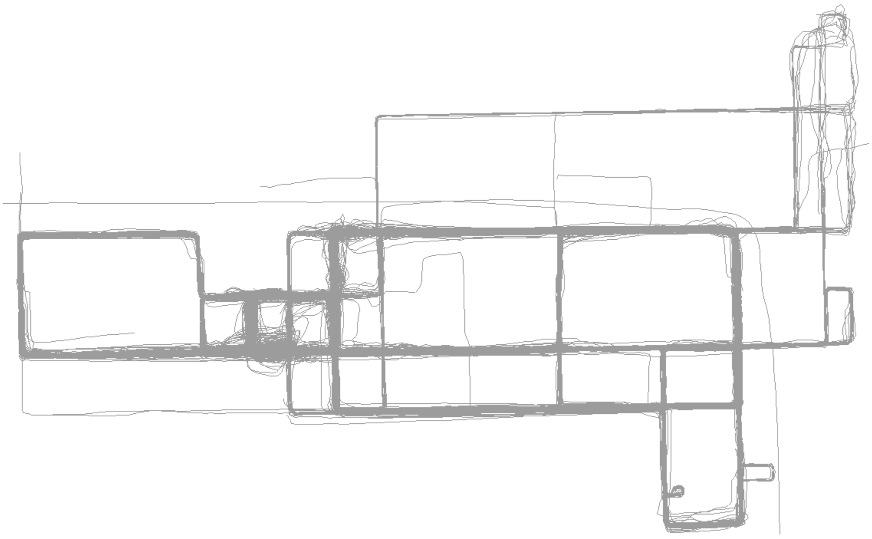


Figure 7.3: Chicago trajectory dataset.

7.1.4 The use of datasets for experiments

In the following, experiments are conducted on different datasets. Table 7.1 shows, which methods were applied to the different datasets. OSM data lack the timestamp information, and the sampling rate may vary significantly. Thus the speed and heading related parameters cannot be precisely calculated to characterize the patterns of travel modes. As a result, the OSM dataset is not used for testing travel mode classification algorithm. Anomalous behavior detection focused on finding patterns in individual car trajectories, and car trajectories from the Tracer dataset are used to perform experiments.

	Travel mode segmentation and classification	Road map refinement	Anomalous behavior detection
Tracer dataset	✓	✓	✓
OSM dataset	×	✓	×
Chicago dataset	✓	✓	×

Table 7.1: Experiments are performed on different datasets.

7.2 Experimental results of travel mode classification

The Tracer trajectories store the travel mode information that is specified by the commuters while they collect the data. Trace dataset is used to train the classification model and test the statistical appreciation of the proposed travel mode classification methodology.

The transferability of the trained SVMs model enables the presented method to be more reliable and time efficient, since the training dataset is not always available in most areas. Thus, we test the transferability of the classification model derived from training Tracer trajectories with the Chicago dataset, and the promising result is discussed.

7.2.1 Tracer dataset

First stage classification results

Trajectories are firstly segmented to movement segments by identifying stops, and the resulting movement segments are classified into 3 classes: walk, bicycle, and motorized vehicles using the Fuzzy-logic system. The classified segments are then linked up to form sub-trajectories of individual unique travel modes.

After the segmentation, classification and construction procedures, 284 sub-trajectories of individual travel modes are derived. These sub-trajectories are composed from 53 walk trajectories, 19 bicycle trajectories and 212 motorized-vehicle trajectories. The first stage classification results are illustrated in Table 7.2, showing that the majority of sub-trajectories are classified correctly. While comparing the results to the reference data inserted by the users, it is found that 50, 18, and 209 sub-trajectories, respectively, are classified correctly.

Table 7.3 shows the error matrix of the first-stage classification. 3 wrongly classified walk sub-trajectories are identified as bicycles. This happens when pedestrians travel at a relatively high speed compared to normal cases. Incorrect bicycle and motorized vehicle sub-trajectories are misclassified as each other. This occurs when a bicycle drives at a relatively high speed and motorized vehicles drive at a relatively very low speed. Thus they show similar characteristics that are hard to separate.

Travel mode	Total	Correct	Wrong	Statistical certainty (%)
Walk	53	50	3	94.3%
Bicycle	19	18	1	94.7%
Motorized vehicle	212	209	3	98.6%
Total	284	277	7	97.5%

Table 7.2: Classification results for the first stage

	Walk	Bicycle	Motorized vehicle
Walk	-	3	0
Bicycle	0	-	1
Motorized vehicle	0	3	-

Table 7.3: Error matrix of first-stage classification

SVM classification results

192 out of 209 sub-trajectories are used for the second stage classification, composed of all travel mode types exist in the motorized vehicle class. 17 sub-trajectories are not used because they are too short having only one movement segment (where trajectories with at least 2 movement segments are used for the SVM classification). These sub-trajectories are divided to training data (87 sub-trajectories) and testing data (105 sub-trajectories). 11 attributes are calculated and applied in the SVM to train the data and get optimized C and γ for the predicting model.

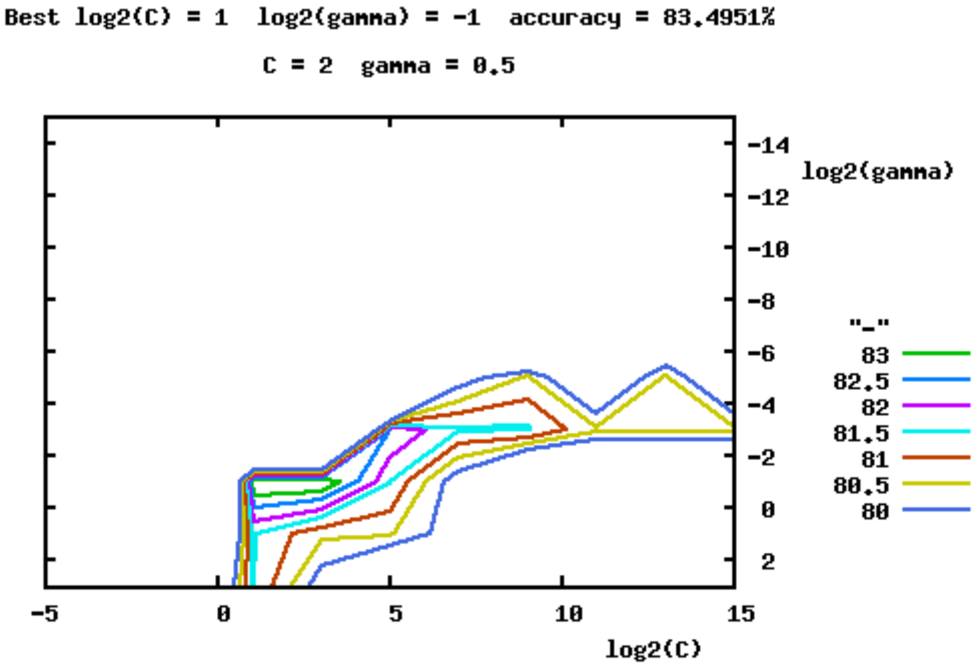


Figure 7.4: Optimized parameters $\mathrm{C} = 2$, $\gamma = 0.5$ and accuracy contour. Grid-search on $\mathrm{C} = 2^{-5}, 2^{0}, \ldots, 2^{15}$ and $\gamma = 2^{-14}, 2^{-12}, \ldots, 2^{2}$ and 5-folder cross-validation on training data. Accuracy contour are in colored lines.

In the training session, grid search is applied on $C = 2^{-5}, 2^{0}, \ldots, 2^{15}$ and $\gamma = 2^{-14}, 2^{-12}, \ldots, 2^{2}$ and for each (C, γ) 5 folder cross-validation is used to identify good regularization parameter C and the kernel parameter γ for the prediction model. As depicted in Figure 3.8, the optimized values are $C = 2$ and $\gamma = 0.5$ with the training accuracy is 83.5%. The resulting parameters for the prediction model are not large thus the model does not overfit the training data (cf. 3.1.2).

SVM classification testing results are showed in Table 7.4. The wrongly classified sub-trajectories are analyzed, and showed in Table 7.5 as an error matrix. It is evident that most car travel modes (55 out of 58) are correctly classified. The sub-trajectories wrongly classified are explained by the fact that they have very short stops together with a low number of stops: 2 or 3 only. Though the use of short sub-trajectories in training phase is employed to avoid over-fitting problem, still, the short trajectories might introduce ambiguous features for the SVMs to predict. Train travel mode also showed perfect classification, for the reason that it is distinct from other motorized travel modes to a large degree. 2 bus sub-trajectories are wrongly recognized as tram. Two wrongly classified tram sub-trajectories are classified as car and bus. Bus and tram might have similar movement characteristics, mainly when buses travel at a relatively higher speed in suburb areas whilst distance between two stops are relatively longer, and with less stops at road intersections - when compared to the city center.

Travel mode	Training data	Testing data	Correct	Wrong	Statistical certainty (%)
Car	49	58	55	3	94.8%
Bus	11	17	14	3	82.4%
Tram	19	18	15	3	83.3%
Train	8	12	12	0	100 %
Total	87	105	96	9	91.4%

Table 7.4: Classification results of SVMs classification

	Car	Bus	Tram	Train
Car	-	0	2	1
Bus	0	-	2	0
Tram	1	1	-	
Train	0	0	0	-

Table 7.5: Error matrix of SVMs classification

7.2.2 Chicago trajectory dataset

The transferability of the SVMs model, which is derived by training the Tracer trajectories, is tested using the Chicago trajectory dataset. The transferability would enable the presented method to be more reliable and efficient, since the training dataset has not always been available in most areas.

The Chicago dataset is composed with 889 trajectories, which cover an area of 7km x 4.5km with an average sampling rate of 3.61s. These trajectories are all collected from university shuttles, which respect to a car travel mode other than a bus because of the fact that they don’t have regular stops as normal buses.

Figure 7.5 indicates the travel mode classification result. In the first classification stage, 3 sub-trajectories (0.3% of total trajectories) are detected as walk and 34 (3.8%) are identified as bicycle. The other 852 trajectories (95.8%) are all correctly classified as motorized-vehicles. 334 trajectories have more than 2 movement segments and are further classified using SVMs method.

Wrongly classified walk trajectories result from low travelling speed, for the rea son of rush-hour traffic jam etc. The trajectories that are classified as bicycle travel mode have relatively low average speed and together with high acceleration.

We apply the same classification model which is trained using the Trace dataset to classify the travel modes of the sub-trajectories derived in the first stage classification. Only 1 trajectory is wrongly detected as bus and 333 trajectories out total 334 trajectories are correctly identified as car travel mode. This bus trajectory has more stops at road intersections compared to other car trajectories.

Overall, classification and transferability results of SVM classification show quite promising results.

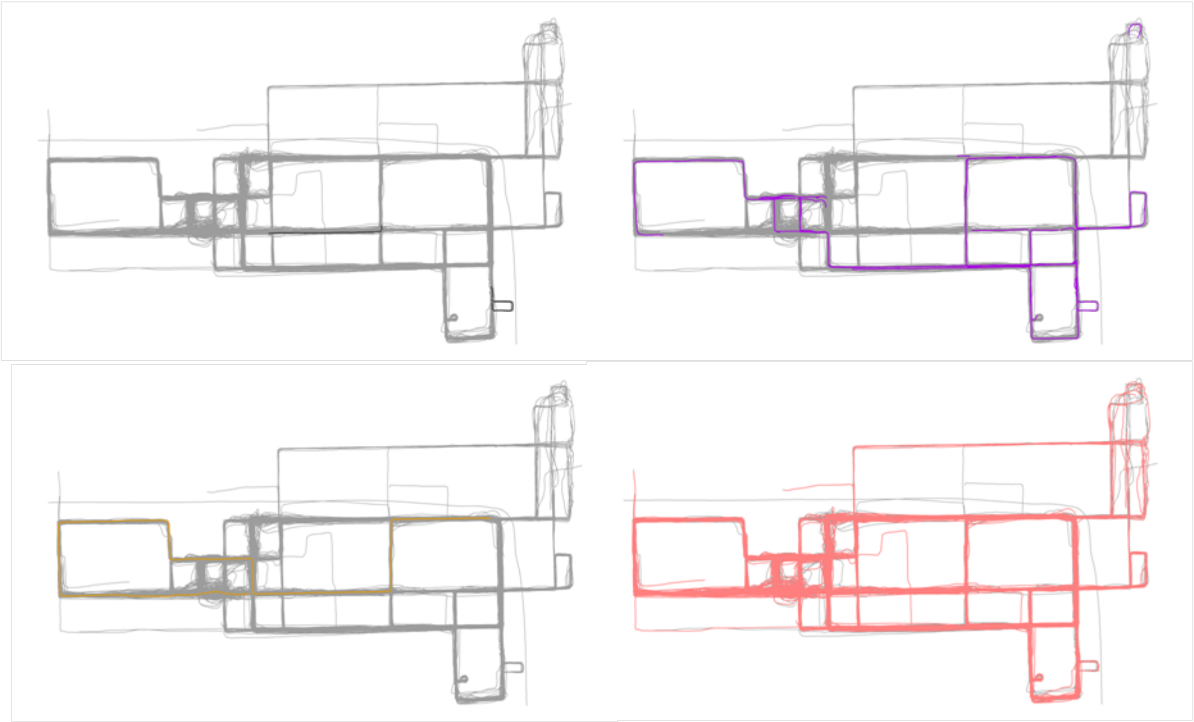


Figure 7.5: Travel mode classification result of Chicago dataset. Walk trajectories are presented in black, bicycle in purple, bus in yellow and car in pink.

7.2.3 Discussion

The algorithm depends on a number of parameters that characterize the patterns of different travel. modes. Thus, a high sampling rate and a good positional accuracy of trajectories are required. The algorithm cannot be applied on OSM dataset, which lacks the information of time, and the sampling rate is uncertain. The SVM classification cannot be implemented on short trajectories that have only one movement segment. As a result, not all trajectories of the dataset can be classified. For example, 334 out of 889 trajectories in Chicago dataset are classified using SVM method.

Among the six travel modes, vehicle travel mode are more likely to be misclassified as bicycle, due to the fact that when vehicles travel at relatively low speed, resulting from traffic jams etc., they show similar characteristics as bicycles and are hard to separate. Car and train travel modes show perfect classification, for the reason that they are distinct from other motorized travel modes to a large degree.

The test of the transferability of the classification method shows promising result. This on one hand proves that the threshold setting is not sensitive to different datasets, and on the other hand ensures that the trained SVM model can be applied on other dataset. The transferability enables the presented method to be more reliable and efficient.

7.3 Experimental results of road map refinement

Experiments are carried out to test the presented method of road map generation using three datasets. Since the OSM trajectory data lack the information of time stamps, speed and heading related parameters cannot be precisely calculated. Thus, the travel mode classification method is not applied on the dataset. Since these trajectories are mainly derived from highways, we assume that they are all vehicular trajectories, and trajectories are matched with their corresponding roads using only geometrical and clustering methods. As for Chicago dataset, 333 car trajectories are correctly recognized and these trajectories are used to refine the car route of OSM road map. An area of the constructed road map from Tracer trajectory dataset is shown, where trajectories of three travel modes exist: tram, car and walk.

7.3.1 OSM trajectory dataset

Construction of road centerlines

Experimental result of constructed road centerlines from OSM trajectory dataset is given in Figure 7.6. Resulting roads are presented using different colors according to the number of corresponding trajectories that are used to extract them. It can be observed that more corresponding trajectories for motorways are found than motorway ramp roads. The standard deviation of the center points shows that: in cases where there is only one lane, they are obviously lower than in the multi-lane case. Similarly, they are lower when a large number of GPS-trajectories have been used. Thus, the standard deviation both represents the accuracy of the measurement of the centerline and is an indication of the width of the road.

Figure 7.7 depicted a complex interchange junction, showing that the trajectories can be separated for each road even when many roads are located close to each other. Particularly, as shown in Figure 7.7 (right top), the major highway and its nearby ramp road have similar direction and they can be separated. More corresponding trajectories are found for the major highways that ramp roads. In Figure 7.7, the bottom ramp road is shifted towards the major highway in the derived map, and the new centerline is compatible with its corresponding trajectories. This also shows that the presented algorithm can find corresponding trajectories for an existing road, even though it contains some degree of error. The new centerline for the upper major highway however is less accurate that it is not straight as it should be. Figure 7.7 (right bottom) shows that the resulting road has more precise information about road curvature.

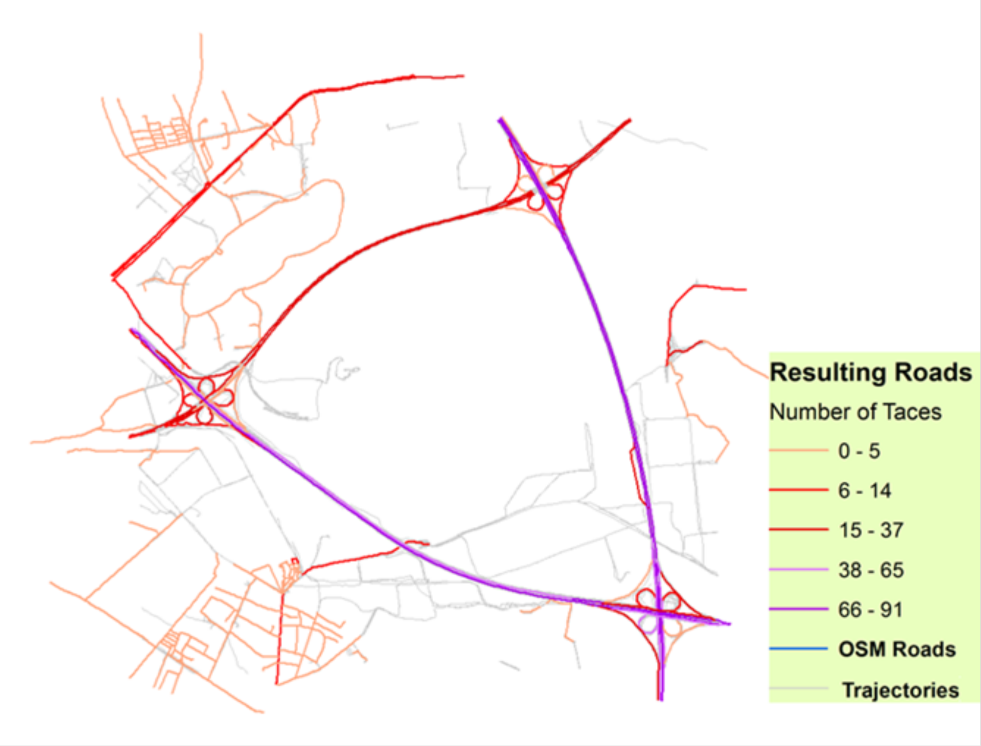


Figure 7.6: Resulting roads constructed from GPS-trajectories.

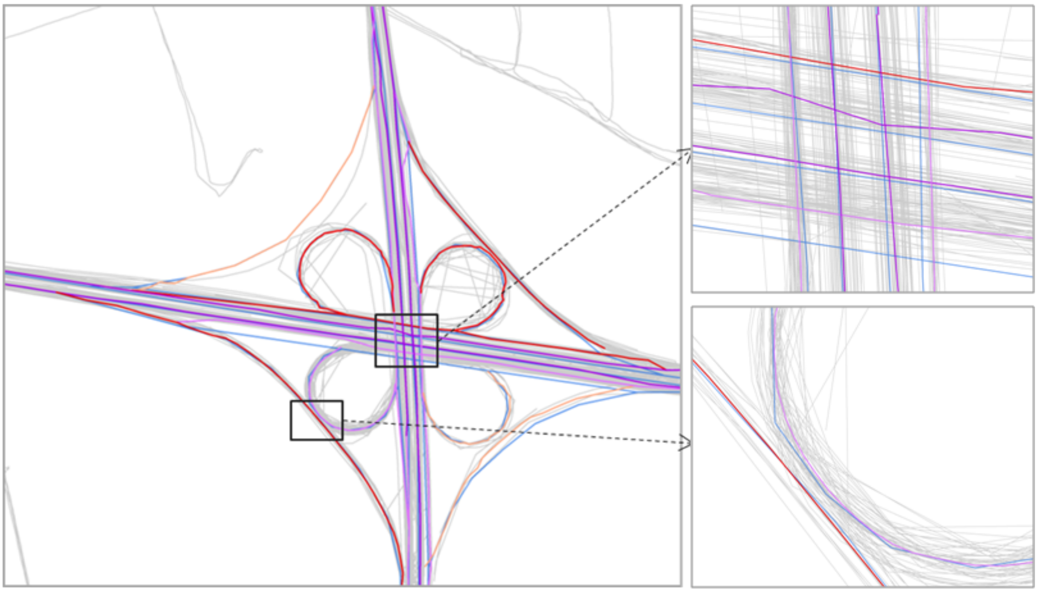


Figure 7.7: Resulting roads of a road junction constructed from GPS-trajectories. The style of the figure follows that of Figure 7.6.

From the left panel of Figure 7.8 we can see that the resulting roads (red) are closer to TeleAtlas data (green), and are consistent with the centerline of the road in image data. The distance that the resulting roads move from the prior roads (blue) can reach 6 meters in some areas. As shown in left panel of Figure 7.8, trajectories are separated and assigned to correct roads even when prior roads are very close to each other. The new road has more precise information about the road curvature.

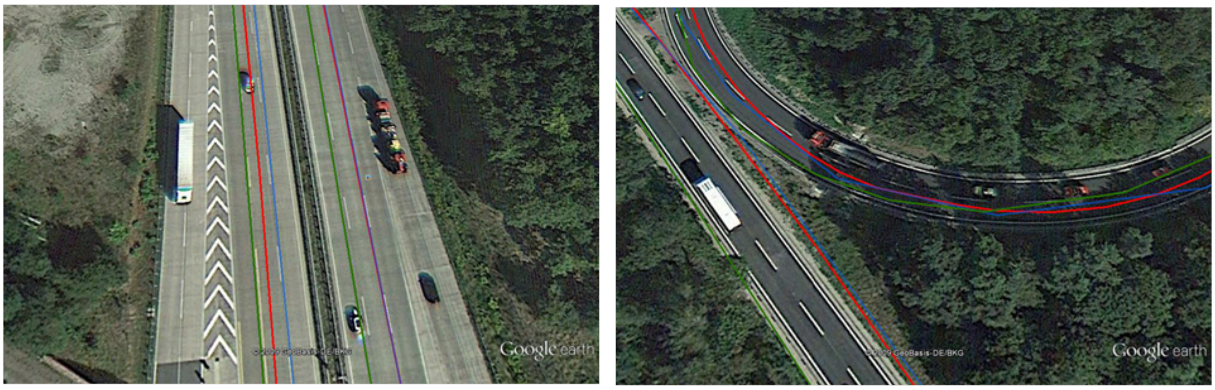


Figure 7.8: Experimental results of the extraction of road centerlines from OSM trajectory dataset. Prior map is shown in blue, and green lines are the TeleAtlas roads. The resulting roads are shown in red. The image data are from Google Earth.

When a road does not have enough sampling GPS trajectories, the reconstruction may be affected by its nearby roads, which have similar direction as it and have more corresponding trajectories. In this situation, the clustering method cannot produce a good result since the concentration of trajectories is too low to form clusters. Figure 7.9 shows two examples of the described situation. The left panel of Figure 7.9 depicts a road intersection, which is shifted towards the road that has more sampling roads. In the right panel of Figure 7.9, a road that has only a few corresponding trajectories is wrongly reconstructed, and the new centerline is located very close to its nearby road that is parallel to it and has more sampling trajectories.

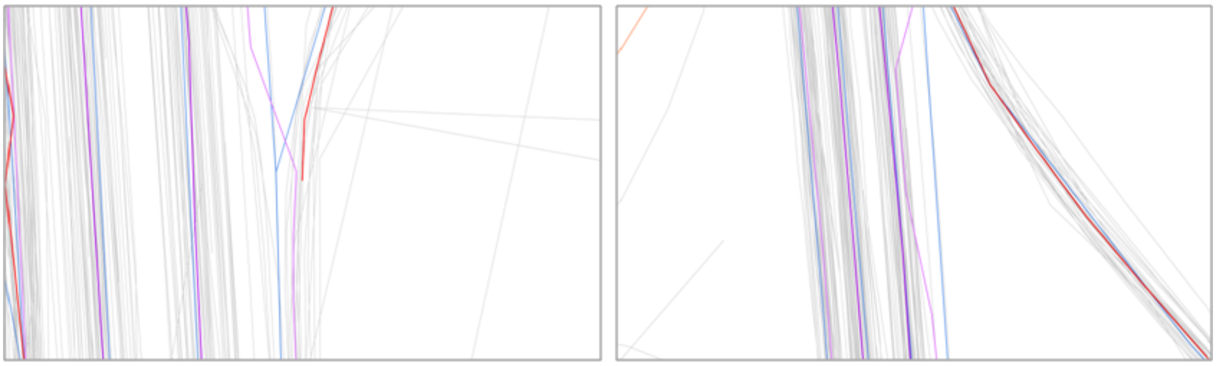


Figure 7.9: Two examples of the resulting roads that are worse than the prior roads. The style of the figure follows that of Figure 7.6.

In order to evaluate the result quantitatively, we compared the result with a standard road map. The standard road map is from TeleAtlas dataset (in GDF-Format); it has an accuracy of 2 to 10 meters. We used a buffer approach as proposed by GOODCHILD and HUNTER (1997), i.e. we evaluate the distance of the prior roads (OSM) and our result from the TeleAtlas data, which are considered have higher positional accuracy. To this end, we split the result roads and the prior roads into line segments, and compare the number of line segments that are completely within 2, 5, 7 meter buffers of the TeleAtlas road map respectively. The result is presented in Table 7.6. In general, the results of our methods fit better to the TeleAtlas dataset than the roads from the OpenStreetMap.

Buffer size (meter)	2	5	7
Result roads	27.4%	61.7%	73.9%
Prior roads (OSM)	14.8%	46.8%	65.8%

Table 7.6: Evaluation of accuracy of the geometric road reconstruction.

Experimental result of estimation of number of lanes

The numbers of lanes are estimated for 42 constructed roads (Figure 7.10). Other roads have only less than 3 sampling trajectories and they are not used in the calculation of numbers of lanes. We compare the result with the true number of lanes, which are derived by investigating the Google image data. For 25 roads out of 42 roads the correct number was determined.

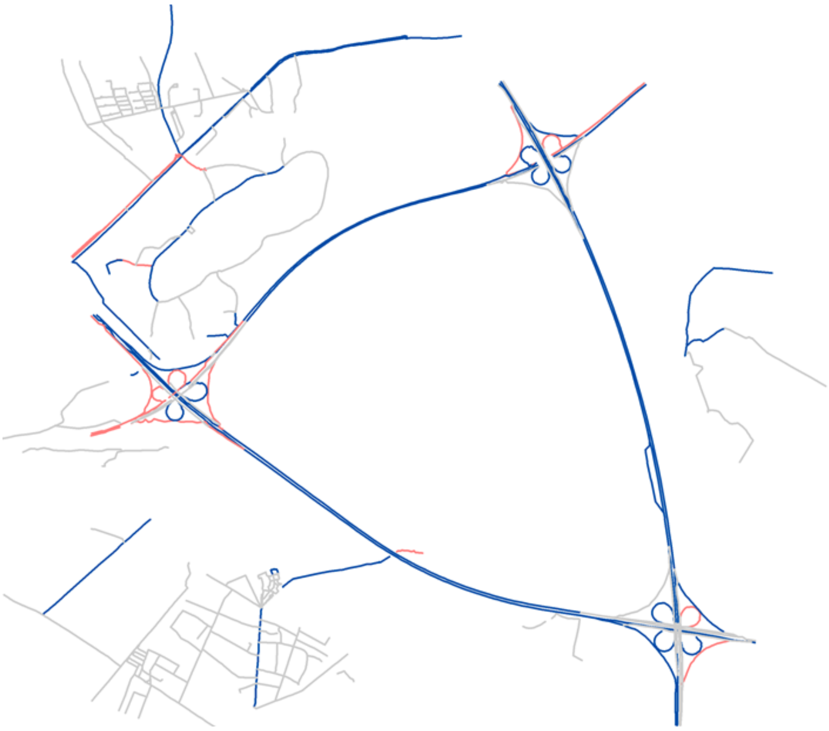


Figure 7.10: Experimental results of estimating the number of lanes. Roads that are given the correct number of lanes are presented in blue and wrongly estimated roads are in red. Roads that have less than 3 roads are presented in grey.

As shown in Figure 7.11, the differences of estimated numbers against true numbers decrease noticeably when the number of trajectories used for the estimation increases. The reason is that, if there are not enough trajectories, the calculated standard deviation of the Gaussian distribution becomes quite larger or smaller than the actual standard deviation. Especially, when the number of trajectories is larger than 16, 13 roads out of 16 roads are given the correct numbers of lanes. The experiment result shows that, if there are more than 16 sampling trajectories, the estimation of number of lanes seems gives satisfy result (see Figure 7.11).

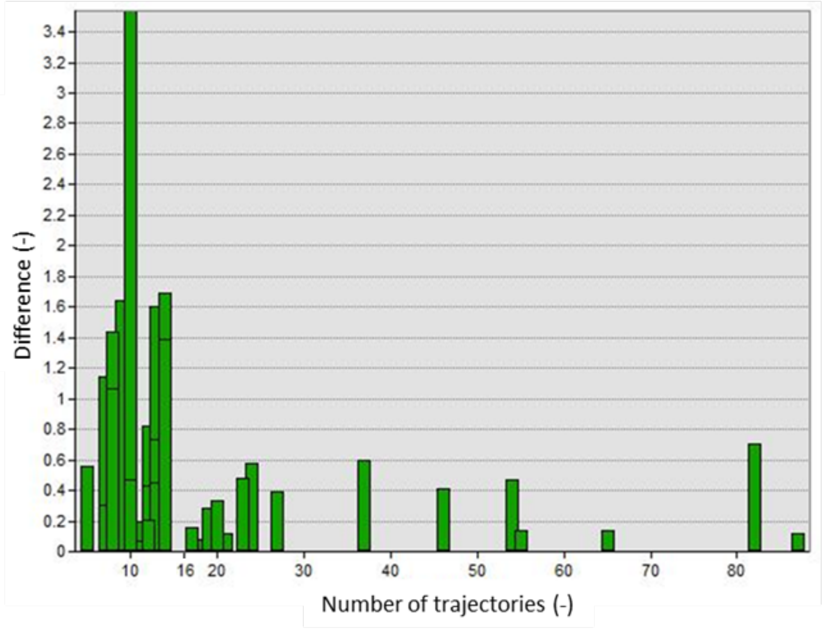


Figure 7.11: Difference between calculated number of lanes and the true number.

7.3.2 Chicago trajectory dataset

After the travel mode classification is processed on Chicago dataset, 333 trajectories are correctly recognized as car travel mode and these trajectories are used to refine car routes of the OSM road map. Thus, only motorways are reconstructed using their corresponding trajectories. Figure 7.12 depicts the experimental results with 94 generated roads. The constructed road centerlines are represented in different colors depending on the number of corresponding trajectories.

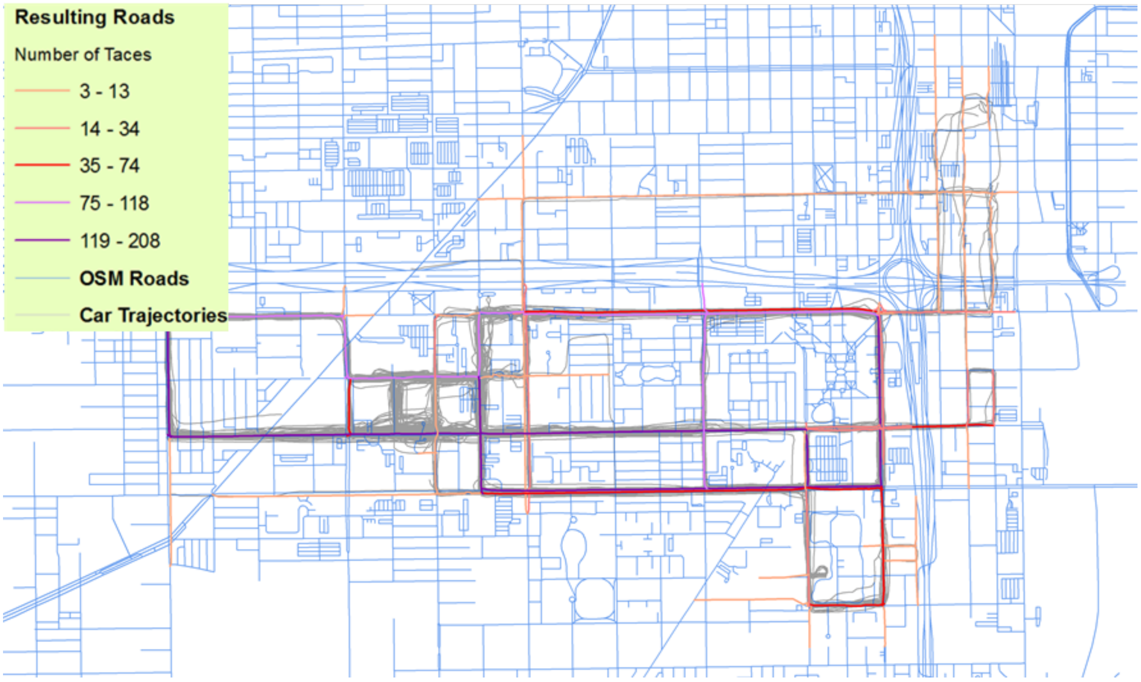


Figure 7.12: Resulting roads constructed from Chicago trajectory data.

Figure 7.13 shows two examples of the constructed roads. With all the trajectories are detected as car trajectories, they are matched to car roads without being confused with nearby roads of other types (i.e. bicycleway, footway, service road, etc.). From Figure 7.13, we can see that the resulting roads are consistent with centerlines of roads in image data. However, the topology of roads is not built and errors may exist at road intersections.

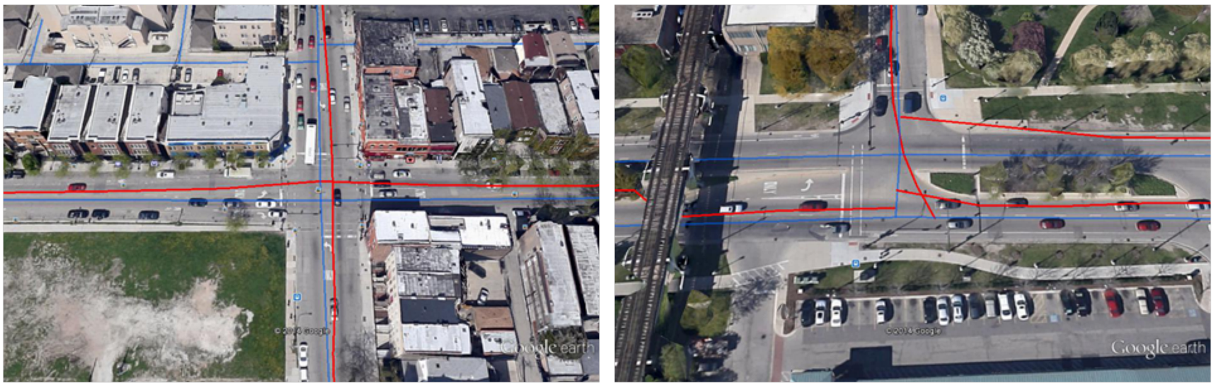


Figure 7.13: Two examples of the constructed road map. Roads are overlaid on Google Earth image data. Prior map is shown in blue, and red lines are the resulting roads.

Figure 7.14 displays the estimated number of lanes for 96 reconstructed roads. After comparing the resulting numbers with the true numbers of lanes that are derived by investigating image data, it is found that 36 out of 96 roads are given the correct numbers. 16 roads have only a few sampling trajectories, and the number of lanes is not estimated for them. 6 one-lane roads are detected, and 32 roads out of 96 roads are identified as two-lane roads. 22 roads are specified as three-lane roads, while 12 and 8 roads are estimated as four-lane and five-lane roads, respectively. Calculated high numbers are mainly from noisier trajectory data. Moreover, a small number of corresponding trajectories together with a high standard deviation indicate that sampling trajectories are widely spread around the road centerline and the estimated number is less reliable.

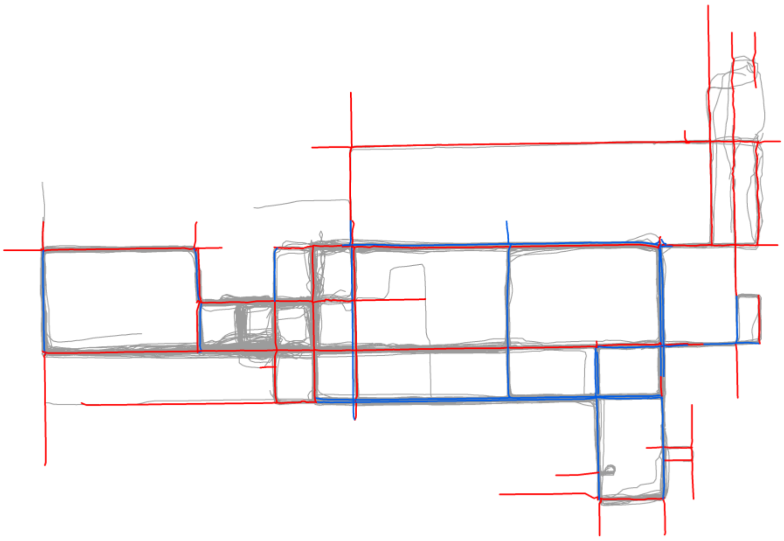


Figure 7.14: The estimated number of lanes for constructed roads. Roads that are given correct numbers are shown in blue and the rest roads are shown in red.

Figure 7.15 displays a road that is not changed although it has many sampling roads. Specifically, the road in Figure 7.15 is a one-way road in the direction from north to south (indicated by the direction of road segments). However the matching result shows that, all the road's corresponding trajectories go in a different direction. As a result, the prior road is not changed. This indicates that the road is given a wrong direction in the prior map (OSM).

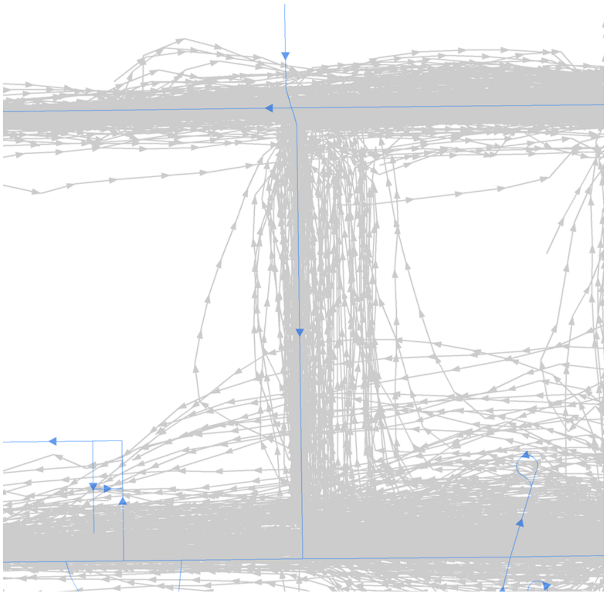


Figure 7.15: A one-way road with a wrong direction in OSM map is detected.

7.3.3 Tracer trajectory dataset

Figure 7.16 shows an area of constructed roads from theTracer trajectory dataset, where trajectories of three travel modes exist: tram, car and walk, which have been identified using the travel mode classification method. This area is chosen because it shows how different types of roads have been reconstructed based on travel modes of their corresponding trajectories. Regardless of the fact that different types of roads are located close to each other and their sampling trajectories are mixed up, they can be separated in the resulting map. In this experiment, service roads (i.e. parking lot roads) are not reconstructed using the car trajectories; however, walk trajectories are corresponding to them.

Tram roads (blue) and some footways are not presented in the OSM map. We use one trajectory as prior information and find the trajectories that are collected from the same road to generate the road centerline. Tram roads and footways along the side of car roads can be separated and appended to the road map. Derived footways are sometimes not very straight, as the GPS data are noisy resulting from low travelling speeds.

For a road in the existing map, sometimes only a part of it has sampling trajectories. In this situation, only the part of the road that has sampling trajectories is reconstructed, whereas the rest part of the road is not updated, since no corresponding trajectories are found for it. This can be seen from several footways, since pedestrians don't strictly walk on footways, especially at road junctions.

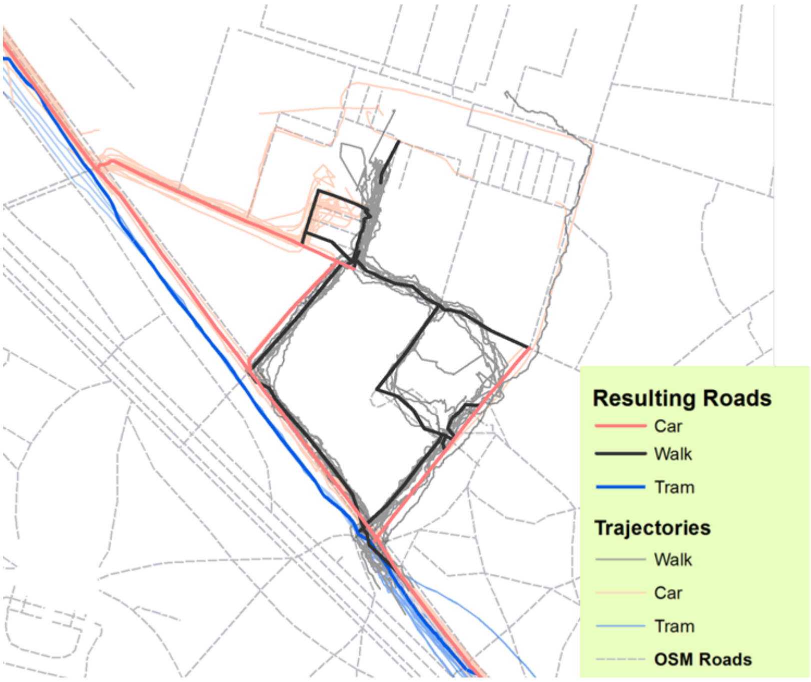


Figure 7.16: The resulting roads that are constructed from the Tracer trajectory dataset.

7.3.4 Discussion

It should be noticed that we focus on the precise matching of prior roads with corresponding trajectories and the reconstruction of road centerlines. The topology is not built for the resulting roads. This mainly affects the road junctions. Post-processing could be conducted to refine the reconstructed roads, e.g. in terms of straightness of roads, reconstruction of junctions, topology, etc.

The algorithm can produce satisfactory results when a large number of trajectories are available for a road. Limited amount of sampling trajectories may lead to a wrong construction, especially when the nearby roads have similar direction as it and have more trajectories than it. This is shown in Figure 7.9.

The walking trajectories sometimes do not correspond to footways, since pedestrians don't consistently use footways. This makes it difficult to construct roads from walking trajectories. When walking trajectories are too far away (i.e., 30 meters) from the possible roads, they cannot be assigned to any roads.

Walking trajectories are much noisier compared with other travel mode trajectories, resulting from low travelling speed. Constructed roads often show many curves other than straight. To overcome this problem, a reduction of errors of the walking trajectories could be performed before the road construction.

Due to the limited number of available trajectories and the high degree of error in GPS data only general information about the number of lanes could be derived. A considerable amount of corresponding trajectories that cluster close to the road may produce a better result. The future work should also include finding exact location of lanes, either by inferring from the centerline and the number of lanes or by clustering trajectories for different lanes.

Figure 7.15 also shows the potential to use the proposed method to detect road restrictions. As the trajectories can be correctly matched to OSM roads, it can detect attribute errors in OSM data (i.e. road restriction, turn restriction, etc.).

7.4 Experiments on anomalous behavior detection

Experiments are performed on 110 car GPS trajectories from the Tracer dataset to test the presented anomalous detection algorithm. The trajectories are collected by the commuters while doing their own business from the city of Hannover. Each trajectory represents a single trip made by the commuter. The presented algorithm is performed to detect anomalous behavior of individual trajectories.

7.4.1 Results of anomalous driving behaviour detection

The presented algorithm is performed on 110 individual trajectories. As displayed in Figure 7.17, a limited portion of these trajectories are recognized as anomalous. Figure 7.18 shows the statistical analysis of the experimental results. 46 out of total 110 trajectories, 3809 points out of total 78653 (4.8%) are detected with a probability of anomaly over 50%. Setting the threshold for the probability of an anomaly to 70%, the number of trajectories decreases noticeably to 24 and the number of involved points drops to 2605 (3.3% of the total number). Anomalous behavior, with a probability of over 80%, is detected in 21 trajectories with 1808 points (2.3% of the total number). There is a very slight decrease in the number of trajectories with more than 85% and 95% probabilities of being anomalous, 19 and 18 respectively; whereas the numbers of involved points drop to 1628 (2.1% of the total number) and 1197 (1.5% of the total number). It can be seen from the statistical analysis of the experimental results, that when the trajectory shows a more than 85% probability of being anomalous, it is quite possible that the following acquired points will deliver an even higher probability of being anomaly. It can be recommended that in practice we can use 85% probability as the threshold to identify anomalous behavior.

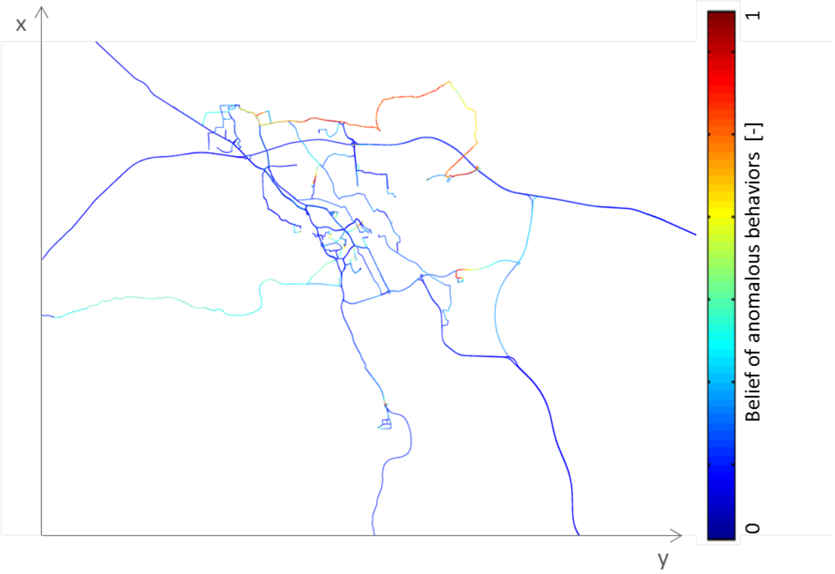


Figure 7.17: Experiment on 110 car trajectories. The belief of anomaly is presented by color of the trajectory.

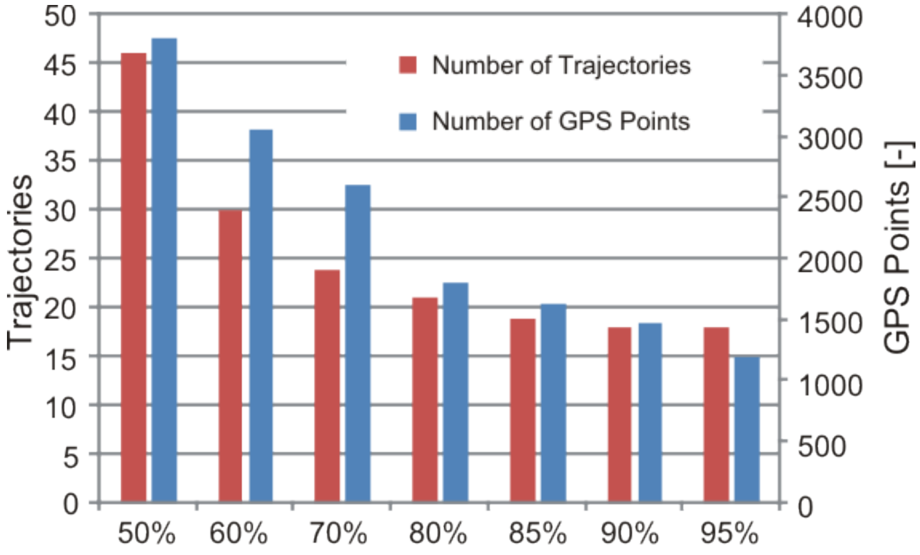


Figure 7.18: Statistical result of detected anomaly as a function of anomaly probability.

Analyses on several typical individual trajectories are given with the following figures.

Figure 7.19 demonstrates a trajectory which is correctly recognized as normal driving, i.e., no obvious anomalous patterns are found. The belief distribution function shows robustness with a maximum belief of 23% and some slight fluctuates, even though the trajectory also contains a few large turns.

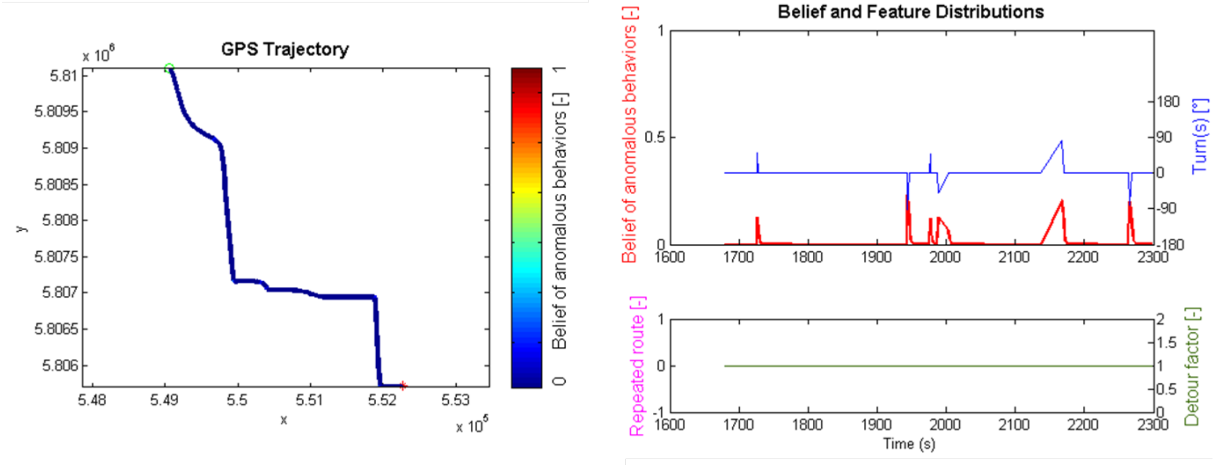


Figure 7.19: An example of a normal driving: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right).

In comparison with the simulated detour case (cf. Figure 6.6), Figure 7.20 shows the detour detection on an actual trajectory.

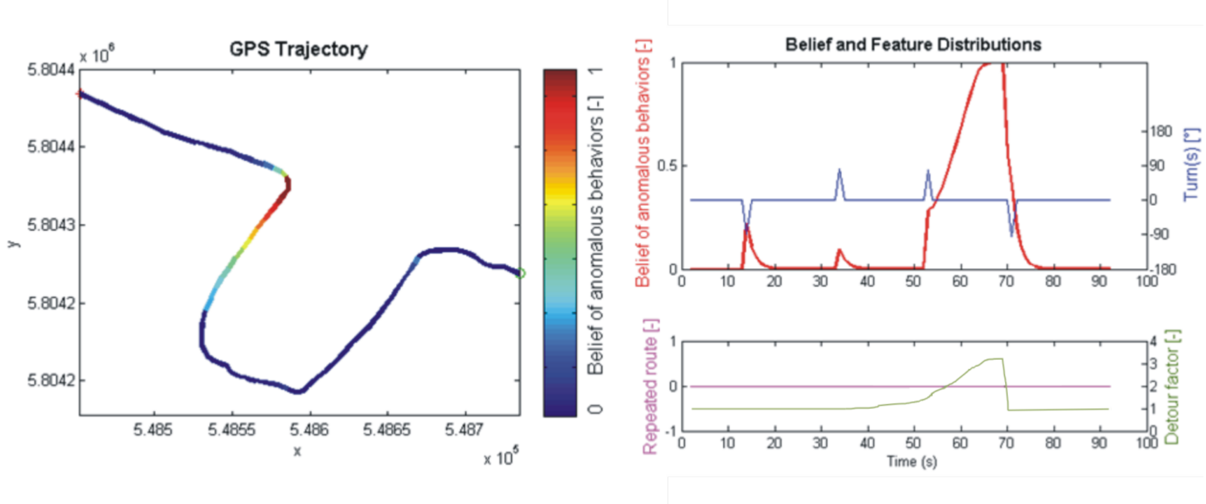


Figure 7.20: An example of a detour: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right).

A more complicated case with many turns, loop and detours is given in Figure 7.21. As shown in the belief and feature distributions (right panel of Figure 7.21), the influence of the first two turns decreases rapidly along the driving and presents actually a normal segment. Forming a loop is in contrast a prominent anomalous behavior, which consists of three sequential right turns and route repetition when the vehicle gets back in the former road. Thus the belief of being anomalous increases dramatically and reaches 100%. After the route repetition, another right turn is detected and the trajectory is still in a detour with a large detour factor and a high belief of anomaly. Although the following trajectory segment is still in the trend of going backwards, resulting from several small right turns, the detour factor keeps decreasing and the belief of being anomalous drops as well.

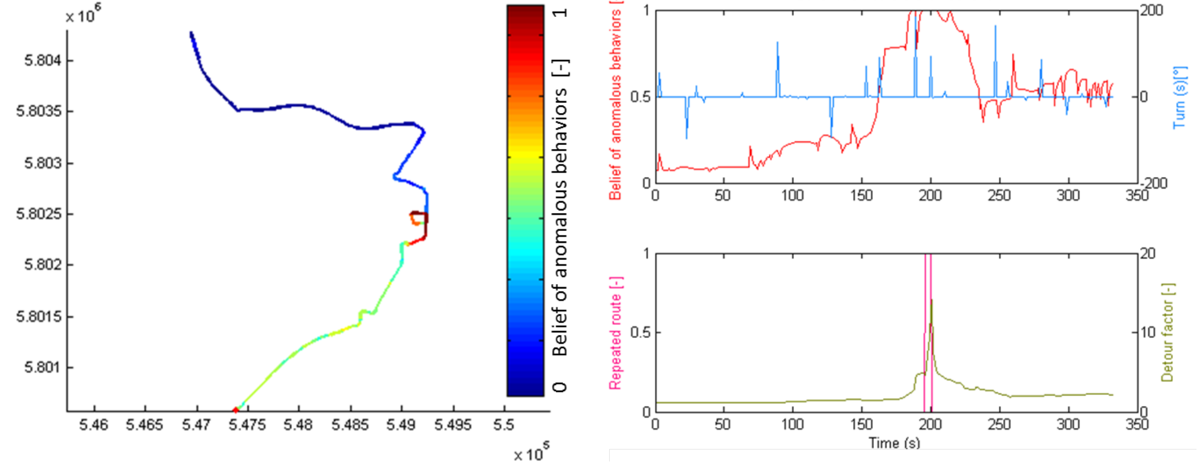


Figure 7.21: A single trajectory with turns, loop and detour: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right).

The above detected anomalous behavior is relatively short term behavior, while as one trajectory is detected as anomalous for a long period as shown in Figure 7.22. The trajectory shows an anomalous driving behaviour since the driver takes a big detour and the detour factor keeps a high level for a long period. The anomalous behavior of the trajectory indicates that the driver may need some kind of navigational instruction (e.g. interesting places, route suggestions).

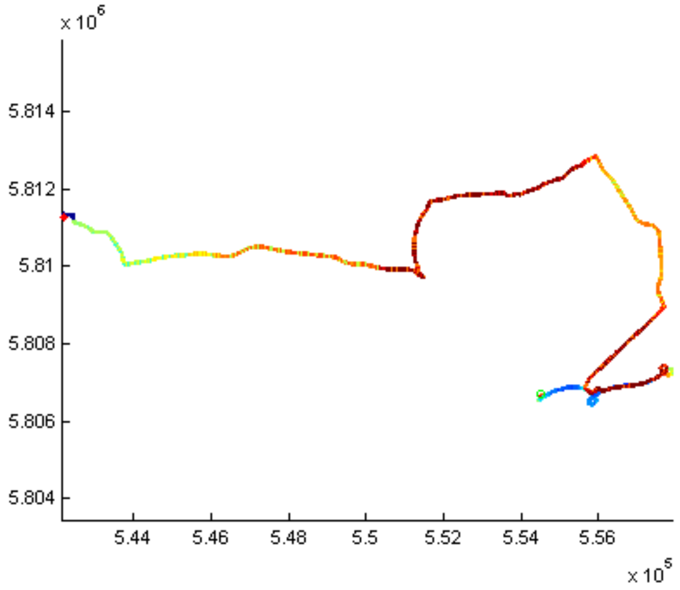


Figure 7.22: A trajectory is detected as anomalous for a long period.

7.4.2 Analysis of collective behavior

Collective travel mode change behavior

Figure 7.23 depicts a scenario of the collective behaviour detection: blue trajectories represent car travel mode and red trajectories represent walking travel mode (right). The left image depicts the default route plan produced via Google Maps, while the right image displays the route which would be taken if the parking lot was available in the data set. By knowing the parking place associated with the desired address the driver is directed to a different but more appropriate and optimized location other than straight to the address, represented by a pink buffer in both images. This scenario shows that the optimized route is much shorter. Not only, that the route is shorter, thus saving driving time, the driver will most probably save some time and frustration in finding this specific parking place.

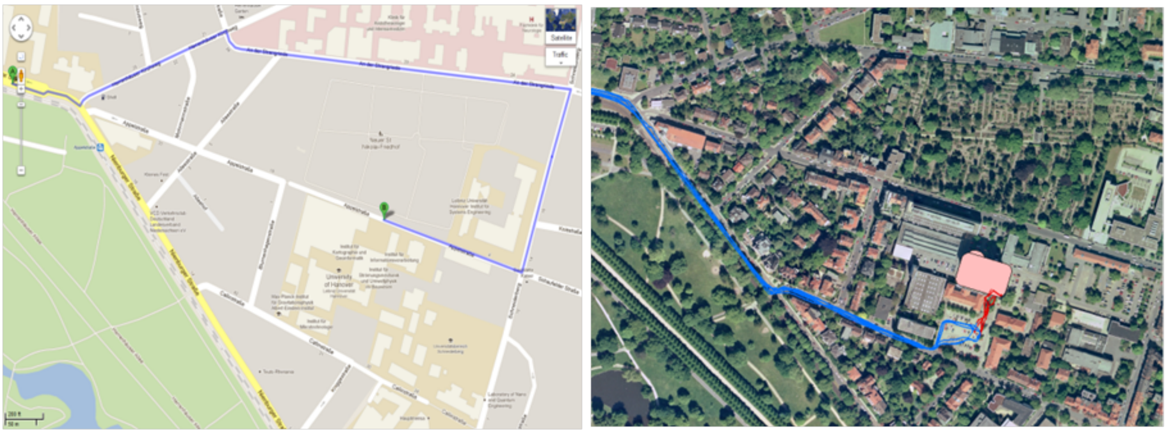


Figure 7.23: Route planning scenario: “off-the-shelf” Google Maps route (left) and optimized route (right). Extraction of parking place associated with desired address avoiding detour. Car trajectories are presented in blue, walking trajectories in red, destination in pink polygon.

The example amplifies the argument: by including the knowledge regarding parking places in the vicinity of building and facilities addresses into the navigation maps and databases, it is possible to construct a more fine-tuned route that answers the user requirements: the closest location where he or she can park the car in order to get to the desired address.

Collective anomalous behavior

Figure 7.24 shows an example in Hannover, Germany, where has a temporary road-block as well as a blind alley nearby (right). Anomalous patterns can be found from multiple sides of the road-block. As shown in trajectories (left), driver 1 from the north saw the sign of road-block and leaved the road to avoid the block and get back to the road again after the block. Driver 2 missed the sign and had to make a U-turn right before the road-block and then performed a detour like driver 1 did to get on with the same direction. Driver 3 from the south turned around even earlier possibly because of the sign or a traffic jam before the crossing. Although the blind alley on the west side is not a temporary setup, it may cause U-turns for the reason that the drivers may be not familiar with this neighbourhood.

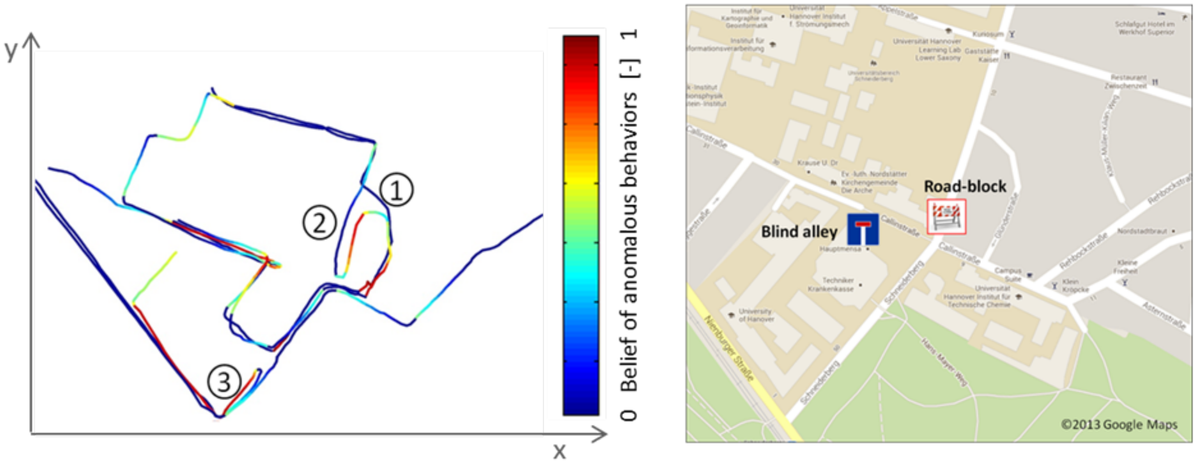


Figure 7.24: Collective anomalous behavior detected in a group of trajectories (left) and the street map (Google Map) that is labelled with the locations of the road-block and the blind alley (right).

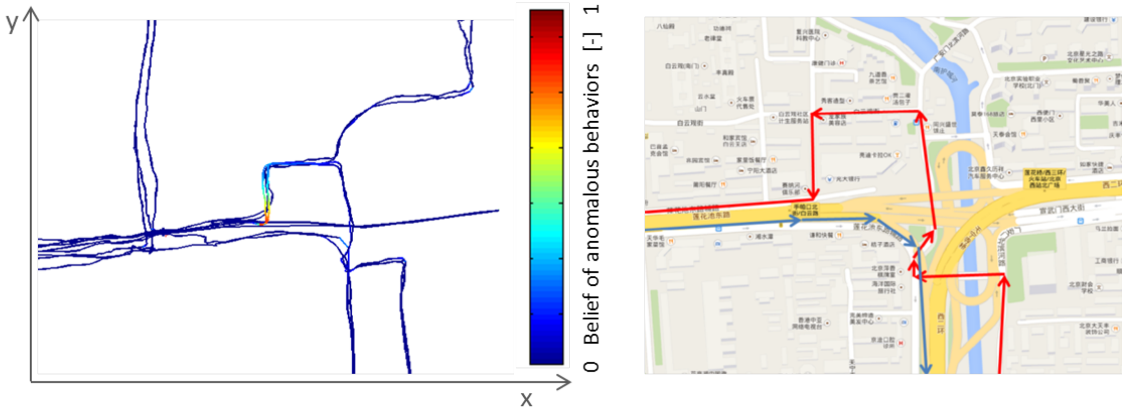


Figure 7.25: Collective anomalous behaviors detected on a collection of trajectories showing road restriction (left) and the Google street map (right).

An experiment on the „GeoLife GPS Trajectories" (ZHENG et. al., 2008; 2009; 2010) at a road junction in Beijing, China shows the potential to detect traffic restrictions using the collective anomalous behavior detected in a group of trajectories. As shown by Figure 7.25, the trajectories going from the bottom to the left show coincident detour while the reversed (from the left to the bottom) trajectories have no anomaly detected. We assume that such phenomena may indicate a potential left turn restriction, which is proven by the street map shown in Figure 7.25 (right panel), i.e., no left turn is possible here because of the cloverleaf junction and the direction restrictions of the streets.

7.4.3 Discussion

The driving behavior is considered as a sequence of turns made by the driver. Thus, the anomalous behavior is characterized by frequent turns and their combination. A detour is detected when sequential same direction turns occur. The detour factor is then calculated to present the degree of the detour. However, if a long straight road is taken before the detour, a small value of the detour factor will be derived. Thus, the possibility of anomalous will increase slowly and cannot deliver a sufficient decision of being anomalous in time. Figure 7.26 illustrates such a situation.

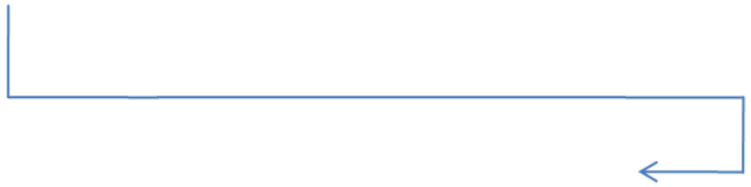


Figure 7.26: An example of a detour that cannot be well characterized by detour factor.

The detour factor for a point is calculated based on the whole history of the trajectory before it. This is to prevent the failure of large detour detection. However, this also introduces a problem that the anomaly sometimes decays very slowly. An example can be seen from Figure 7.22. A scaling scheme may be applied to tackle the problem by using only the recent trajectory segment to infer the detour factor.

Examples of collective behavior detection are discussed. Further implements of automatic detection should be conducted according to the method described in Chpater 6.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

Integrating massive crowd-sourced GPS-trajectory data with the OSM road map relies on the precise matching between the trajectories and existing road segments. The precise matching has to tackle the error both in the GPS data and OSM data. Therefore, geometrical and fuzzy c-mean clustering methods are used to separate trajectories. In addition, matching according to the travel mode of the trajectory is essential. Since different types of roads are often located closely to each other and the sampling trajectories of them may overlap and not be separated, identifying correctly the road type from which a GPS-trajectory is collected is important for the implementation of matching it to the corresponding road in reference map.

We developed a novel method towards improvement of existing OSM road data from incoming, massive amounts of GPS-trajectory data. Travel modes of trajectories can be precisely identified and such information is used to match the noisy crowd-sourced trajectories with correct reference roads, from which they are collected. Furthermore, the availability of travel mode information provides the potential to construct roads of specific types (e.g., bicycle roads and pedestrian roads). It is also presented that the proposed method can take one trajectory as a reference road and use other incoming trajectories to generate the road centerline. Besides road centerlines, attribute information e.g. number of lanes is derived through mining GPS-trajectory data.

A multi-stage method towards the automatic segmentation and classification of travel modes from GPS-trajectories is presented. Movement segments of GPS-trajectories and sub-trajectories comprised of individual travel modes are found with very high certainty. New parameters are introduced, other than the commonly used speed related parameters, namely heading and pattern recognition classifiers, which proved to produce reliable results, contributing to the complete classification process introduced. A SVMs supervised learning method is used to automatically classify sub-trajectories with high statistical certainty. Also, this research introduces the capacity of classifying six different travel modes (bus travel mode is separated from car travel mode) as opposed to the common five; thus, expanding the potential of the classification process and introducing new capacities.

Anomalous driving behavior cannot be measured directly. Thus, the driving behavior is presented as a sequence of control steps rather than as a sequence of raw positions and velocities. The control steps are extracted from the trajectory of the vehicle as long-term features (turn, repetition and detour). These long-term features are then remodelled using Markov chain and used as inputs for a variant of recursive Bayesian filter to deliver an optimal probability distribution of the potential anomalous driving behavior over time.

Detected anomalous behavior from trajectory indicates where the driver is likely meeting navigational problem and an assistant is required. Thus, a dynamic inference process is necessary to detect anomalous behavior from individual trajectories. A recursive belief filter is conducted for the dynamic and unsupervised detection of anomalous patterns.

Collective behavior of a group of trajectories is analysed to reveal the underlying information. Group travel mode change behavior can be used for the extraction of parking places associated with a specific address. This information can be used by the route planning scheme to enable the construction of a more fine-tuned route that will prevent subsequent detours for finding parking places. Collective anomalous behavior of a group of trajectories shows a potential of reflecting traffic issues, e.g., complicated crossings, unexpected blind alleys and temporary road-blocks.

8.2 Future Works

There are a number of issues which could be followed up based on the research of this thesis.

The first issue relates to the reconstruction of the roads: the main focus of this thesis was the identification of contributing trajectories and reconstruction of road centrelines. The topology of the resulting roads is not constructed. Post-processing should be investigated to refine the reconstructed roads, e.g. in terms of straightness of roads, the structure of junctions, topology, etc.

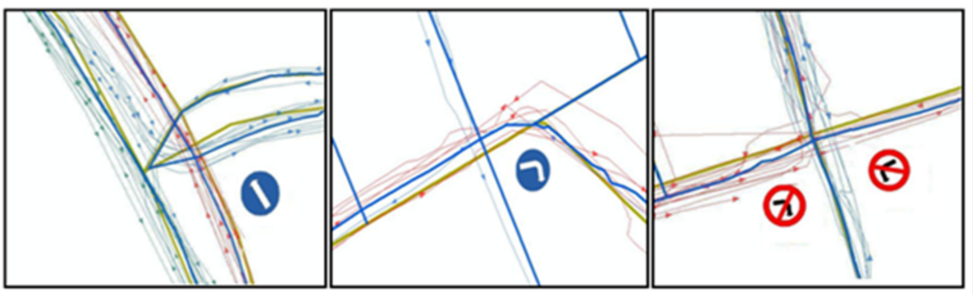
The presented method of travel mode classification extracts the characteristics of different travelling behavior using speed, heading and movement pattern related parameters. It is expected that the trajectories have a sampling rate that is not lower than several seconds and the degree of noise in GPS data should not be too high. Otherwise the calculated parameters will be inaccurate and cannot deliver a sufficient representation of the behavior. Future work should investigate the possibility of identifying travel modes from trajectories with low sampling frequency.

The integration approach should be extended to better compensate for changed prior information, e.g. OSM map. In the thesis, we have shown that, when a road is not presented in the reference map, the process may use one trajectory as the prior information to start the construction. Thus missing roads may be appended to the prior map. Future work may include automatically detect the changed area from trajectories data. Furthermore, also investigations with respect to structural changes of the prior information has to be included, e.g. the fact that a new lane is built: the process has to take possible and plausible changes into account and allow these structural changes as soon as enough when current information votes for it.

The future work should also include finding the exact location of lanes based on the extracted information of road centerline and the number of lanes. Especially the lane structure of road junctions may be appended to the road map. This can be achieved by inferring from the centerline and the number of lanes or by clustering trajectories that have a high positional accuracy.

Since the trajectories can be precisely matched to the prior roads, a quality evaluation of the navigation maps, such as OSM, TomTom, HERE, etc., may be conducted. Besides the positional accuracy, various attribute information could be assessed by comparing the travel modes, directions of trajectories with road types, road restrictions information in the mentioned navigation datasets.

Besides the number of lanes, road restriction information can also be mined when large amount of trajectories for a road are available. When all trajectories of one road show identical turn behavior at a specific junction, turn restriction may be inferred from such collective behavior. Figure 8.1 shows three examples of road restriction (only straight on, only right turn, no right turn and no left turn) that are extracted from trajectories.



(1) Only_straight_on (2) Only_right_turn (3) No_right_turn, and No_left_turn

Figure 8.1: Examples of turn restrictions. GPS trajectories for different roads are shown in different colors with arrows indicating their direction. Brown lines are prior roads and reconstructed road centerlines are represented by blue lines.

Typical driving behavior, such as looking for parking places, finding interesting places, etc. are interesting for navigational service and location based service. This typical driving behavior can be defined more accurately when introducing other information, including surrounding interesting places, parking availability, action pattern of the driver, etc.

Bibliography

AHMED M., KARAGIORGOU S., PFOSER D., and WENK C., 2014. A Comparison and Evaluation of Map Construction Algorithms. Arxiv.org pre-print.

BOHTE, W., and MAAT, K., 2009. Deriving and Validating Trip Purposes and Travel Modes For Multi-Day GPS-Based Travel Surveys: A Large-Scale Application in The Netherlands. Transportation Research Part C: Emerging Technologies, 17(3), pp. 285-297.

BURGES C., 1998. A Tutorial On Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery. Kluwer Academic Publishers, Boston, Vol.2.

CAO H., MAMOULIS N., and CHEUNG D., 2005. Mining Frequent Spatiotemporal Sequential Patterns. In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM ’05), Houston, Texas, pp: 82-90.

CAO, L. and KRUMM J., 2009: From GPS Traces to a Routable Road Map, 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), November 4-6, 2009, Seattle, WA, pp. 3-12.

CHANG, C., and LIN, C., 2011. LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

CHEN, C., CHENG, Y., 2008. Roads digital map generation with multi-track GPS data. Proc. Workshops on Education Technology and Training, and on Geoscience and Remote Sensing, pp. 508-511.

CHEN, Y. and KRUMM J., 2010. Probabilistic Modeling of Traffic Lanes from GPS Traces, Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information System, New York, pp. 81-88.

CHUNG, E., and SHALABY, A., 2007. A Trip Reconstruction Tool for GPS-based Personal Travel Surveys. Transportation Planning and Technology, 28(5), pp. 381-401.

DAVIES, J.J., BERESFORD, A.R., HOPPER, A., 2006. Scalable, Distributed, Real-time Map Generation. IEEE Pervasive Computing 5(4), pp. 47-54.

DEAN T. and KANAZAWA K., 1989. A model for reasoning about persistence and causation. Computational Intelligence, vol. 5(3), pp. 142-150.

DIARD J., BESSIÈRE P., and MAZER E., 2003. A survey of probabilistic models, using the bayesian programming methodology as a unifying framework. In *The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*, Singapore.

FATHI A., KRUMM J., 2010. Detecting Road Intersections from GPS Traces. Geographic Information Science Lecture Notes in Computer Science, Vol. 6292, pp. 56-69.

FEUERHAKE U., KUNTZSCH C. and SESTER M., 2011. Finding Interesting Places and Characteristic Patterns in Spatio-Temporal Trajectories, Proceedings of the 8th International Symposium on Location-Based Services, Vienna.

GHAHRAMANI Z., 2001., An Introduction to Hidden Markov Models and Bayesian Networks, Hidden Markov Models. Applications in Computer Vision, World Scientific Publishing Co., Inc., River Edge, NJ.

GIANNOTTI F., NANNI M., PINELLI P., and PEDRESCHI D., 2007. Trajectory Pattern Mining. In Berkhin P, Caruana R, and Wu X (eds) KDD ’07: Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, ACM Press, pp. 330–39

GONG, H., CHEN, C., BIALOSTOZKY, E. and LAWSON, C.T., 2011. A GPS/GIS Method for Travel Mode Detection in New York City. Computers, Environment and Urban Systems (in press).

GOODCHILD, M. F. and Hunter, G. J., 1997. A simple positional accuracy measure for linear features. International Journal of Geographical Information Science. 11(3), pp. 299-306.

GOODCHILD, M.F., 2007. Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. International Journal of Spatial Data Infrastructures Research, 2007, Vol. 2, pp. 24-32.

GRIFFITHS, T.L. and YUILLE, A., 2006. A primer on probabilistic inference, Trends in Cognitive Sciences Supplement to special issue on Probabilistic Models of Cognition, vol. 10(7), pp. 1–11.

GUO, D., 2008: Mining Traffic Condition from Trajectories, Fifth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 4, pp.256-260.

HAKLAY, M. and WEBER P., 2008. OpenStreetMap: User-Generated Street Maps. IEEE Pervasive Computing.

HSU C., CHANG, C., and LIN, C., 2003. A Practical Guide to Support Vector Classification. Technical Report Department of Computer Science and Information Engineering, National Taiwan University.

MACDONALD I., ZUCCHINI W., 1997. Hidden Markov and Other Models for Discrete-valued Time Series, Chapman & Hall, London.

MEYER, D., LEISCH, F., and HORNIK, K., 2003. The Support Vector Machine under Test. Neurocomputing

55(1–2), pp. 169–186.

MITROVIC D., 2005. Reliable Method for Driving Events Recognition. IEEE Trans. Intell. Transp. Syst.,vol. 6, no. 2, pp.198 -205.

MURPHY, K., 1998. A Brief Introduction to Graphical Models and Bayesian Networks. http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html, accessed 2014.

OLIVEIRA, M., TROPED, P., WOLF, J., MATTHEWW, C. and CROMLEY, E., 2006. Mode and Activity Identification Using GPS and Accelerometer Data. Proceedings of Transportation Research Board 85th Annual Meeting, 12p.

OPENSTREETMAP. http://wiki.openstreetmap.org/wiki/Stats#Users_and_GPX_uploads, assessed 2014.

PATROUMPAS K., 2013. Multi-scale Window Specification over Streaming Trajectories. Journal of spatial information science, Numver 7, pp. 45-75.

PATTERSON, D. J., LIAO, L., FOX, D. and KAUTZ H., 2003. Inferring High-Level Behavior from Low-Level Sensors. International Conference on Ubiquitous Computing (UbiComp), 2003.

PEARL J., 1988. Probabilistic Reasoning in Intelligent Systems :Networks of Plausible Inference. Morgan Kaufmann, San Mateo, Ca.

PENTLAND A., LIU A, 1999. Modeling and Prediction of Human Behavior. Neural Computation, Vol. 11, pp. 229-242.

PENTLAND A., LIU A., 1999. Modeling and Prediction of Human Behavior, Neural Computation, vol. 11, pp. 229-242.

RABINER L. R.1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE Trans. on ASSP, Vol. 77(2),pp. 257–285.

RAFTERY A., 1985. A model for high-order Markov chains, J.R. Statist. Soc. B 47; pp. 528–539.

REDDY, S., BURKE, J., ESTRIN, D, HANSEN, M., and SRIVASTAVA, M., 2008. Determining Transportation Mode on Mobile Phones, 12th IEEE International Symposium on Wearable Computers, 2008. ISWC 2008., pp. 25-28.

SÄRKKÄ S., 2013. Bayesian Filtering and Smoothing. Cambridge University Press.

SATHYANARAYANA A., BOYRAZ P., HANSEN J. H.L., 2008. Driver Behavior Analysis and Route Recognition by Hidden Markov Models. Proc. IEEE Intl. Conf. on Vehicular Electronics and Safety, pp. 276-281.

SAYDA, F., 2005: Involving LBS users in data acquisition and update, Proceedings of the AGILE, Portugal.

SCHROEDL, S., WAGSTAFF, K., ROGERS, S., LANGLEY, P. and WILSON C., 2004.Mining GPS Traces for Map Refinement. Data Mining and Knowledge Discovery, 2004, 9(1): pp. 59-87.

SCHÜSSLER, N. and AXHAUSEN, K. W., 2008. Processing GPS Raw Data without Additional Information. Working paper, 15p.

SESTER M., FEUERHAKE U., KUNTZSCH C. and ZHANG L., 2012. Revealing Underlying Structure and Behaviour from Movement Data, KI - Künstliche Intelligenz, pp. 1-9.

SESTER, M., 2009. Cooperative Boundary Detection in a Geosensor Network using a SOM, ICC Chile, 2009.

SMOLA, A. J. and SCHÖLKOPF, B., 1998.On a Kernel–based Method for Pattern Recognition, Regression, Approximation and Operator Inversion, Algorithmica, pp. 211231.

SOLEYMANI A., CACHAT J., ROBINSON K., DODGE S., KALUEFF A.and WEIBE R., 2014. Integrating cross-scale analysis in the spatial and temporal domains for classification of behavioral movement. Journal of spatial information science, number 8, pp. 1-25.

STOPHER, P. R., JIANG Q., and FITZGERLAD, C., 2005. Processing GPS Data from Travel Survey. Proceedings of 2nd Int. Colloquium on the Behavioral Foundations of Integrated Land-use and Transportation Models: Frameworks, Models and Applications. Toronto, June 2005.

TORKKOLA K., VENKATESAN S., LIU H., 2005. Sensor Sequence Modeling for Driving. Flairs Conference, Clearwater Beach, Florida, USA, pp. 721-727.

TROPED, P. J., OLIVEIRA, M. S., MATTHEWS, C. E., CROOMLEY, E. K., MELLY, S. J., and, CRAIG, B. A., 2008. Prediction of Activity Mode with Global Positioning System and Accelerometer Data. Medicine & Science in Sports & Exercise, 40(5), pp 972-978.

TSUI, S. A., and SHALABY, A. S., 2006. Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems. Journal of the Transportation Research Board, Issue 1972, pp. 38 – 45.

VAPNIK V., 1995. The Nature of Statistical Learning Theory. Springer, N.Y.. ISBN0-387-94559-8.

VAPNIK V., GOLOWICH S., and SMOLA A., 1997. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. Advances in Neural Information Processing Systems, vol. 9, pp. 281– 287.

WIKIPEDIA. Fuzzy-c-means Clustering. http://en.wikipedia.org/wiki/Cluster_analysis#-Fuzzy_c-means_clustering, accessed 2014.

WOLF, J., 2006. Applications of New Technologies in Travel Surveys. Travel Survey Methods - Quality and Future Directions, pp. 531–544, Elsevier, Oxford.

XU, C. JI, M., CHEN W., and ZHANG Z., 2010. Identifying Travel Mode from GPS Trajectories through Fuzzy Pattern Recognition. 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol. 2, pp.889 – 893.

ZHANG, L., THIEMANN, F. and SESTER, M., 2010. Integration of GPS Traces with Road Map, Proceedings of the Workshop on Computational Transportation Science in conjunction with ACM SIGSPATIAL, San Jose, USA, 2010

ZHENG Y., LI Q., CHEN Y., XIE X., MA W., 2008. Understanding Mobility Based on GPS Data. In Proceedings of ACM conference on Ubiquitous Computing (UbiComp), Seoul, Korea. ACM Press: 312-321.

ZHENG Y., XIE X., MA W., 2010. GeoLife: A Collaborative Social Networking Service among User, location and trajectory. Invited paper, in IEEE Data Engineering Bulletin. 33, 2, pp. 32-40.

ZHENG Y., ZHANG L., XIE X., MA W.,2009. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of International conference on World Wild Web (WWW), Madrid Spain. ACM Press, pp. 791-800.

ZHENG, Y., LIU, L., WANG, L. and XIE, X., 2008. Learning Transportation Mode from Raw Gps Data for Geographic Applications On The Web. Proceedings of the 17th international conference on World Wide Web, WWW '08, pp. 247 – 256.

ZOU X. and LEVISON D., 2006. Modeling Intersection Driving Behaviors: A Hidden Markov Model Approach (I). Transportation Research Record Journal of the Transportation Research, pp.16-23.

List of Figures

Figure 2.1: Graph of the dramatic growth in terms of registered users and contributed GPS data to OSM on a yearly basis (OPENSTREETMAP, 2014)........ 8

Figure 2.2: The JOSM interface........ 9

Figure 2.3: ROGERS et al. (1999) refine a road map by averaging it with a DGPS trajectory (1999)........ 9

Figure 2.4: EDELKAMP and SCHRÖDL (2003) inference road map from trajectories using K-mean algorithm........ 10

Figure 2.5: CAO and KRUMM (2009) propose a physical attraction model, which enable the displacement of individual GPS points........ 11

Figure 3.1: The SVM learns an optimal model which best classifies the two classes. Red dots have a label $\mathbf{yi} = +\mathbf{1}$ while blue dots have a label $\boldsymbol{yi} = -\mathbf{1}$........ 18

Figure 3.2: An overfitting classifier (left) and a better classifier (right) (HSU et al., 2003). The decision boundary between negative examples (red circles) and positive examples (blue crosses) is shown as a thicker line. The lighter lines are on the margin........ 20

Figure 3.3: The effect of the Gaussian kernel parameter $\boldsymbol{\gamma}$ with a given value of $\mathbf{C}$ (HSU et al., 2003). The figure style follows that of Figure 3.2........ 20

Figure 3.4: A directed aryclic graph represents the conditional independence relations in $\mathrm{P(W, X, Y, Z,)}$........ 22

Figure 3.5: A Bayesian Network presenting a first-order Markov process........ 23

Figure 3.6: A Bayesian Network presenting the conditional independence relations of a hidden Markov model........ 24

Figure 4.1: An example of a trajectory that is composed of 3 sub-trajectories: train (blue), walk (black) and bus (yellow)........ 28

Figure 4.2: Speed pattern of movement segments and stops (top) and corresponding heading changes magnitude (bottom). GPS-trajectory representing approximately 15 minutes of travelling is divided into different individual movement segments. Car movement segments are in blue, walk segments are in red, and identified stops in black........ 30

Figure 4.3: Speed pattern of an example trajectory with stops between two adjacent movements are identified. Movement segments are presented in blue and stops are in red........ 32

Figure 4.4: Typical speed patterns of different travel mode movement segments........ 33

Figure 4.5: Value range of mean speed (top), maximum speed (middle), and mean heading changes (bottom) for different travel modes........ 34

Figure 4.6: The example trajectory after performing the first stage classification. Each segment is identified as vehicle (blue) or walk (black). Stops are represented in red........ 35

Figure 4.7: Sub-trajectory construction result of the example trajectory. 3 vehicle sub-trajectories are identified and are represented in blue, green and yellow. Walks are shown in black and stops in red. 36

Figure 4.8: Vehicle Sub-trajectory travel modes are identified using SVMs method. Train travel mode is presented in blue, bus travel mode in yellow, walking in black and stops in red. 39

Figure 5.1: Workflow for extracting of road centerline. 42

Figure 5.2: GPS trajectories, before and after preprocessing. 43

Figure 5.3: The process of extracting road centerlines. 44

Figure 5.4: Geometrical conditions for matching corresponding trajectories to a prior road. 44

Figure 5.5: Getting candidate trajectory set for the road using profiles perpendicular (in blue) to prior road's centerline (in red). 45

Figure 5.6: An example where the two parallel roads are closely located. The estimated centerlines (in blue) cannot be separated using geometrical matching method. 46

Figure 5.7: Standard deviation σ as a measure for the width of the road. 48

Figure 5.8: The distribution of GPS trajectories of a road can be modeled as a Gaussian distribution. 49

Figure 6.1: Turn extraction: the accumulated heading changes are extracted as a turn feature at the last time stamp. 52

Figure 6.2: Two special cases showing the turn extraction. The left part of the figure shows the error in the data and the heading changes are ignored. The right panel shows a circular road situation where heading changes continuously and a left turn is extracted. 53

Figure 6.3: Detour detection and detour factors calculation. A detour is detected when the heading changes accumulate to more than $135°$. Start and end points of the detours are represented as red and green circles respectively. Detour factors are calculated by the ratio of the length of the trajectory and the direct distance between the current point and the start point (green dashed line). 54

Figure 6.4: A complicated detour example: a small detour $\mathrm{T}[\mathrm{p}8, \mathrm{p}16]$ is a part of a large detour $\mathrm{T}[\mathrm{p}3, \mathrm{p}16]$. The detour factors for the small detour are calculated twice according to the two start points. Start and end points of the detours are represented as red and green circles respectively. 55

Figure 6.5: The Markov chain integrating long-term features. 56

Figure 6.6: Simulated trajectory of detour (left) with start position (green circle), end position (red asterisk) and the belief of anomalous behavior shown with scaled color. Three high-level features: Turns (blue), repeated route (magenta) and detour factor (green) are plotted together with the belief of an anomaly over time (right). 58

Figure 6.7: Simulated trajectory of repetition: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right). 59

Figure 6.8: Building buffer containing walking travel modes to/from car parking place: car trajectories are represented in blue, walk trajectories are represented in red. 61

Figure 7.1: Graphical User Interface of the Tracer App: main view (left); travel mode selection (right). 64

Figure 7.2: The Tracer trajectory dataset: 192 GPS-trajectories were collected in the study-area of Hannover City. 65

Figure 7.3: Chicago trajectory dataset. 66

Figure 7.4: Optimized parameters $\mathrm{C} = 2, \gamma = 0.5$ and accuracy contour. Grid-search on $\mathrm{C} = 2 - 5, 20, \ldots, 215$ and $\gamma = 2 - 14, 2 - 12, \ldots, 22$ and 5-folder cross-validation on training data. Accuracy contour are in colored lines. 68

Figure 7.5: Travel mode classification result of Chicago dataset. Walk trajectories are presented in black, bicycle in purple, bus in yellow and car in pink. 70

Figure 7.6: Resulting roads constructed from GPS-trajectories. 72

Figure 7.7: Resulting roads of a road junction constructed from GPS-trajectories. The style of the figure follows that of Figure 7.6. 72

Figure 7.8: Experimental results of the extraction of road centerlines from OSM trajectory dataset. Prior map is shown in blue, and green lines are the TeleAtlas roads. The resulting roads are shown in red. The image data are from Google Earth. 73

Figure 7.9: Two examples of the resulting roads that are worse than the prior roads. The style of the figure follows that of Figure 7.6. 73

Figure 7.10: Experimental results of estimating the number of lanes. Roads that are given the correct number of lanes are presented in blue and wrongly estimated roads are in red. Roads that have less than 3 roads are presented in grey. 74

Figure 7.11: Difference between calculated number of lanes and the true number. 75

Figure 7.12: Resulting roads constructed from Chicago trajectory data. 75

Figure 7.13: Two examples of the constructed road map. Roads are overlaid on Google Earth image data. Prior map is shown in blue, and red lines are the resulting roads. 76

Figure 7.14: The estimated number of lanes for constructed roads. Roads that are given correct numbers are shown in blue and the rest roads are shown in red. 76

Figure 7.15: A one-way road with a wrong direction in OSM map is detected. 77

Figure 7.16: The resulting roads that are constructed from the Tracer trajectory dataset. 78

Figure 7.17: Experiment on 110 car trajectories. The belief of anomaly is presented by color of the trajectory. 79

Figure 7.18: Statistical result of detected anomaly as a function of anomaly probability. 80

Figure 7.19: An example of a normal driving: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right). 80

Figure 7.20: An example of a detour: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right). 81

Figure 7.21: A single trajectory with turns, loop and detour: trajectory with colors indicating the belief of anomaly (left) and the distributions of the belief and behavior features over time (right). 81

Figure 7.22: A trajectory is detected as anomalous for a long period. 82

Figure 7.23: Route planning scenario: “off-the-shelf” Google Maps route (left) and optimized route (right). Extraction of parking place associated with desired address avoiding detour. Car trajectories are presented in blue, walking trajectories in red, destination in pink polygon. 82

Figure 7.24: Collective anomalous behavior detected in a group of trajectories (left) and the street map (Google Map) that is labelled with the locations of the road-block and the blind alley (right). 83

Figure 7.25: Collective anomalous behaviors detected on a collection of trajectories showing road restriction (left) and the Google street map (right). 83

Figure 7.26: An example of a detour that cannot be well characterized by detour factor. 84

Figure 8.1: Examples of turn restrictions. GPS trajectories for different roads are shown in different colors with arrows indicating their direction. Brown lines are prior roads and reconstructed road centerlines are represented by blue lines. 87

List of Tables

Table 7.1: Experiments are performed on different datasets.......... 66

Table 7.2: Classification results for the first stage 67

Table 7.3: Error matrix of first-stage classification 68

Table 7.4: Classification results of SVMs classification 69

Table 7.5: Error matrix of SVMs classification.......... 69

Table 7.6: Evaluation of accuracy of the geometric road reconstruction. 74

Acknowledgments

I would like to thank many people who have assisted me throughout my doctoral studies over the years.

First and foremost, I express my sincerest gratitude to my supervisor, Prof. Dr.-Ing. habil. Monika Sester, for her constant support in making this dissertation possible. Her gift for conceptualization, her enduring encouragement, and her practical advice have been a consistent source of support for me throughout my PhD. One simply could not wish for a better or friendlier supervisor.

The studies discussed in this thesis would not have been possible without the financial support of China Scholarship Council (CSC). I also give my gratitude to the Promotionsabschlussförderung program of the Leibniz University of Hannover, which financially supported me in the last phase of the research process.

I thank all my colleagues at the institute for their support and assistance in many aspects. I specially thank the data integration group members that provide valuable feedback at our meetings. I would like to express my warmest thanks to my colleagues and friends, Dr. Sagi Dalyot and Dr. Hai Huang, who have always given me great help during the years of study.

Words fail me to express my appreciation to my dear mother and my deceased father, for their great role in my life and their sacrifices for me and my siblings. My two sisters and brother have given me continuous moral support and love throughout, for which my mere expression of thanks likewise does not suffice.

Last but not least, I’d like to express my deepest gratitude to my loving husband, whose dedication, love and persistent confidence in me, has taken the load off my shoulder throughout my PhD. Thank you Jian, I would not manage to finish this work without your support. Thank you my cute little daughter, Mia, who kept me smiling during tough times in the PhD pursuit.

Curriculum Vitae

Personal Infomation

Name	Lijuan Zhang
Data of Birth	16. Oct. 1984
Place of Birth	Hebei, P.R. China
Citizenship	Chinese
Marital Status	Married with Jian Li, one child

Education

09/2002 – 07/2006	Bachelor of Science in Surveying and Mapping Engineering, China University of Mining and Technology-Beijing
09/2006 – 07/2009	Master of Science in Cartography and Geomatics, China University of Mining and Technology-Beijing
09/2009 -	Beginning of Ph. D. Studies at Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Under Supervision of Prof. Dr.-Ing. habil. Monika Sester

Experience

10/2005 – 07/2009	Research assistant, National Geomatics Centre of China, Beijing. Prof. Dr. J. Jiang
12/2011 – 04/2014	Part-time Tutor, Institute of Cartography and Geoinformatics, Leibniz Universität Hannover