



HAL
open science

Deep Learning based Vehicular Mobility Models for Intelligent Transportation Systems

Jian Zhang

► **To cite this version:**

Jian Zhang. Deep Learning based Vehicular Mobility Models for Intelligent Transportation Systems. Automatic Control Engineering. Ecole Centrale de Lille, 2018. English. NNT : 2018ECLI0015 . tel-02136219

HAL Id: tel-02136219

<https://theses.hal.science/tel-02136219>

Submitted on 21 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : | 3 | 6 | 7 |

CENTRALE LILLE

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

Spécialité : Automatique, génie informatique, traitement du signal et des images

par

ZHANG Jian

Master of Engineering of Beijing Jiao Tong University (BJTU)

Doctorat délivré par Centrale Lille

Titre de la thèse :

**Modèles de Mobilité de Véhicules par Apprentissage Profond
dans les Systèmes de Transport Intelligents**

Soutenue le 7 décembre 2018 devant le jury :

M. Pierre BORNE	École Centrale de Lille	Président
M. Ismael LOPEZ-JUAREZ	Centre de Recherche CINVESTAV, Mexique	Rapporteur
Mme. Shaoping WANG	Beihang University, Chine	Rapporteur
M. Alejandro CASTILLO ATOCHE	Universidad Autónoma de Yucatán, Mexique	Examineur
M. Abdelkader EL KAMEL	École Centrale de Lille	Directeur de Thèse
M. Gaston LEFRANC	Pontificia Universidad Católica de Valparaíso, Chili	Examineur
Mme. Zhuoyue SONG	Beijing Institute of Technology, Chine	Examineur
Mme. Liming ZHANG	University of Macau, Chine	Examineur

Thèse préparée dans le Centre de Recherche en Informatique, Signal et Automatique de Lille
CRISAL - UMR CNRS 9189 - École Centrale de Lille
École Doctorale Sciences pour l'Ingénieur - 072

Serial N° :

3	6	7
---	---	---

CENTRALE LILLE

THESIS

presented to obtain the degree of

DOCTOR

Topic: Automatic Control, Computer Engineering, Signal and Image Processing

by

ZHANG Jian

Master of Engineering of Beijing Jiao Tong University (BJTU)

Ph.D. awarded by Centrale Lille

Title of the thesis:

**Deep Learning based Vehicular Mobility Models
for Intelligent Transportation Systems**

Defended on December 7, 2018 in presence of the committee:

M. Pierre BORNE	École Centrale de Lille	President
M. Ismael LOPEZ-JUAREZ	Centre de Recherche CINVESTAV, Mexico	Reviewer
Ms. Shaoping WANG	Beihang University, China	Reviewer
M. Alejandro CASTILLO ATOCHE	Universidad Autónoma de Yucatán, Mexico	Examiner
M. Abdelkader EL KAMEL	École Centrale de Lille	Phd Supervisor
M. Gaston LEFRANC	Pontificia Universidad Católica de Valparaíso, Chile	Examiner
Ms. Zhuoyue SONG	Beijing Institute of Technology, China	Examiner
Ms. Liming ZHANG	University of Macau, China	Examiner

Thesis prepared within the Centre de Recherche en Informatique, Signal et Automatique de Lille
CRIStAL - UMR CNRS 9189 - École Centrale de Lille
École Doctorale Sciences pour l'Ingénieur - 072

*To my parents,
to all my family,
to my professors,
and to all my friends.*

Acknowledgements

This research work has been carried out at “Centre de Recherche en Informatique, Signal et Automatique de Lille (CRISTAL)” in Centrale Lille, within the research group “Optimisation : Modèles et Applications (OPTIMA)” from September 2015 to December 2018. This work is financially supported by China Scholarship Council (CSC). Without the founding of CSC, I would never have this valuable living and studying experience in France.

First and foremost I offer my sincerest gratitude to my PhD supervisor, Prof. Abdelkader EL KAMEL, for his continuous support of my Ph.D. experience and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better tutor and mentor for my Ph.D. study. He has provided his supervision, valuable guidance, continuous encouragement as well as given me extraordinary experiences throughout my Ph.D. experience.

Besides my supervisor, I would like to thank Prof. Pierre BORNE for his kind acceptance to be the president of my PhD Committee. I would also like to express my sincere gratitude to Prof. Ismael LOPEZ-JUAREZ and Prof. Shaoping WANG, who have kindly accepted the invitation to be reviewers of my PhD thesis, for their encouragement, insightful comments and helpful questions. My gratitude to Prof. Alejandro CASTILLO ATOCHE, Prof. Gaston LEFRANC, Prof. Zhuoyue SONG and Prof. Liming ZHANG, for their kind acceptance to take part in the jury of the PhD defense.

I am also very grateful to the staff in Centrale Lille. Vanessa FLEURY, and Brigitte FONCEZ have helped me in the administrative work. Many thanks go also to Patrick GALLAIS, and Gilles MARGUERITE, for their kind help

ACKNOWLEDGEMENTS

and hospitality. Special thanks go to Christine VION, Martine MOUVAUX for their support in my lodgement life.

My sincere thanks also goes to Dr. Chen XIA, Dr. Bing LIU, Dr. Yihan LIU and Dr. Qi SUN, for offering me the useful advices during my study in the laboratory as well as after their graduation.

I would like to take the opportunity to express my gratitude and to thank my fellow workmates in CRISAL: Qing SHI for the stimulating discussions for the hard teamwork. Also I wish to thank my friends and colleagues: Yue WANG, Ben LI, Yuchen Xie, Paul Cazenave, etc., for all the fun we have had in the past three years. All of them have given me support and encouragement in my thesis work.

All my gratitude goes to Ms. Lucie VERDIERE, my French teacher, who showed us the French language and culture. She inspired me with the French culture, opened my appetite for art, history and enriched my life in France.

A special acknowledgment should be shown to Prof. Wei LU and Prof. Weiwei XING at the School of Software Engineering, Beijing Jiao Tong University, who enlightened me at the first glance of research. I always benefit from the abilities that I obtained on his team.

Last but not least, I convey special acknowledgement to my parents, Chunlai ZHANG and Yuemei CHENG, for supporting me to pursue this degree and to accept my absence for four years of living abroad.

Villeneuve d'Ascq, France
December, 2018

ZHANG Jian

Contents

Acknowledgements	i
Table of Contents	iii
List of Figures	vi
List of Tables	x
List of Algorithms	xii
Abbreviations	xv
General Introduction	1
1 Introduction	7
1.1 Background	7
1.1.1 Motivation	7
1.1.2 Intelligent Transportation Systems	8
1.2 Mobility Models in Vehicular Ad hoc Network	13
1.2.1 Brief introduction to VANET	13
1.2.2 Major concerns of VANET	14
1.2.3 Self-Driving Cars and ACC systems	16
1.3 Challenges, Opportunities and Connections	19
1.3.1 Realistic level: another starting point	19
1.3.2 Traffic prediction with Deep Learning approaches	21
1.3.3 Relationship with Self Driving	22
1.4 Conclusion	22

CONTENTS

2	Neural network based data-driven mobility model	25
2.1	Introduction	25
2.1.1	Neural Network in intelligent traffic	27
2.2	Problem descriptions - Why using NN	28
2.3	Neural network structure and backpropagation algorithm	30
2.4	NN Application in traffic simulation problem	34
2.4.1	Pre-processing the obtained data	35
2.4.2	Deep Learning Method	37
2.4.3	Learning Process	40
2.4.4	Implement simulator for virtual traffic flow	43
2.4.5	Performances	45
2.5	Discussion and Conclusion	51
2.5.1	Discussion	51
2.5.2	Conclusion	52
3	Enhanced Mobility Model with HMM	53
3.1	Introduction	54
3.2	Traditional Mobility Models and their adaptation	54
3.2.1	Mobility Models	55
3.2.1.1	Basic categories on Mobility Models	55
3.2.1.2	Movement restrictions in Mobility Models	57
3.2.1.3	Adaptation to Probabilistic Models	59
3.2.2	Basic adaptation in 1-dimension	62
3.2.3	Adaptation of Car-following model in 2-dimension	69
3.2.4	Summary on Adaptation method of Mobility Models	76
3.3	HMM for Mobility Model improvements	77
3.3.1	HMM in intelligent traffic	80
3.3.2	HMM: Basic knowledge and its denotation	80
3.3.2.1	Definitions and denotations	80
3.3.2.2	The three problems	81
3.3.3	HMM combined Mobility Model	87
3.3.3.1	Scaling Problem	92
3.3.3.2	Multi-observation sequences	93
3.3.3.3	Partial knowledge estimation	94
3.3.4	Performance and result	95

3.4	Discussion and Conclusion	99
3.4.1	Discussion	99
3.4.2	Conclusion	99
4	Experiment platform and scenario simulation	101
4.1	Introduction	101
4.2	UML design of NN-VMM	102
4.2.1	Commercial usage	105
4.2.2	Open-source platform	106
4.3	Scenario simulation	108
4.3.1	Highway scenario reproduction	108
4.3.2	Scenario extension: MIXED NN-VMM/TP-AIM	110
4.3.3	Applications in Transportation technology	116
4.4	Discussion and Conclusion	118
	Conclusions and Perspectives	123
	Résumé Étendu en Français	129
	References	139

CONTENTS

List of Figures

1.1	Vehicles per thousand people: US (over time) and other countries (in 2006 and 2016) (Source http://www.greencarcongress.com/2018/10/20181010-fotw.html)	9
1.2	Sensors planted in Luxembourg area (Source https://vehilux.gforge.uni.lu/index.html)	12
1.3	Instance for road ITS system layout	13
1.4	Vehicular Ad hoc Networks (Source johndayautomotiveelectronics.com)	15
1.5	Stanley at Grand Challenge 2005	17
1.6	Sensors on Google's self-driving car (Source www.aventurine.com)	18
2.1	Comparison on average speed of vehicle flows with different models	28
2.2	Demonstration of the architecture of Neural Network	31
2.3	Demonstration of a single neuron	31
2.4	Overview of the proposed method	34
2.5	A single data query formed to represent the vehicles' state	35
2.6	Definitions of different areas and the relevant distances	36
2.7	Comparison on different Batch Size	39
2.8	Structure of a recurrent neural network	41
2.9	Structure of a LSTM network	41
2.10	Composition of a LSTM unit and its denotations	42
2.11	Detail System Design in UML Class Diagram	44
2.12	Vehicle distribution and vehicle flow composition	48

LIST OF FIGURES

2.13	Neural network training, result presented by tensor-board, which indicates the changes on cost through iterations	49
2.14	Output with Gaussian noise in velocity, acceleration and distance	50
2.15	A simulation for sudden stop scenario	50
3.1	Categorization on Mobility Models: scale	55
3.2	Category Overlapping on Mobility Models	56
3.3	Flow chart of the adaptation algorithm	61
3.4	Adaptation of representation of a model	62
3.5	Schematic diagram on the location of V_i , V_j , and possible evaluation interval of v'	66
3.6	Overlapping area demonstration for different $[v'_{min}, v'_{max}]$	68
3.7	"Velocity"- "Spacing" plane in 2-dimension and probability denotations	72
3.8	Impacts on different speed of preceding vehicle	74
3.9	Differences of the flow chart between enhanced method and the original one	79
3.10	Dependency and composition relationship between different components in the enhanced method	79
3.11	Transition and Emission of our HMM	88
3.12	Histogram of marginal and joint distribution: (Left) Marginal distribution of Speed; (Middle) Marginal distribution of Spacing; (Right) Joint distribution of Speed and Spacing	90
3.13	Convergence of Baum Welch algorithm for multi-observation sequences, measured by Logarithm-Probability	96
3.14	Result of Learned NN after HMM labeling by LM-algorithm	98
3.15	Result of Learned NN after HMM labeling by BFGS-algorithm	98
4.1	Flow Chart of the simulation system NN-VMM	103
4.2	Sequence Diagram of the simulation system	104
4.3	State chart Diagram of a vehicle object	105
4.4	User interface of PTV Studio:PTV Vissim	105
4.5	Interface TraCI and its relation with SUMO	106
4.6	Neural network construction on Tensorflow: open-source platform for deep learning	107

LIST OF FIGURES

4.7	Our method in different layer	108
4.8	Road structure of the section that surveillance camera covers	109
4.9	Highway reproduction scenario presented by SUMO	109
4.10	Performance comparison from different models: original observed data, NN-VMM, Enhanced NN-VMM and Car-following Model	111
4.11	the MIXED simulation system: NN-VMM/TP-AIM	112
4.12	Road infrastructure of the under simulated intersection	115
4.13	Impacts on throughput of different composition of the traffic flow	115
4.14	SUMO presentation of experiments on NN-VMM/TP-AIM	116
4.15	Highway structure of this simulation scenario	117
4.16	Queue's forming over time	118
4.17	Traffic simulation for queue length estimation by SUMO	119
1	Véhicules par millier d'habitants: États-Unis et autres pays (en 2006 et 2016) (Source http://www.greencarcongress.com/2018/10/20181010-fotw.html)	130
2	Conception de système dans un diagramme de classes UML	133
3	L'organigramme de la simulation avec modèle de mobilité améliorée	135
4	Procédure de l'algorithme d'adaptation	136

LIST OF FIGURES

List of Tables

3.1	Adaptation result for a demonstration	75
3.2	Joint and marginal distribution of Speed and Spacing	90
3.3	Deviation on frequency/probability of corresponding events . .	91
3.4	Estimated Transition Matrix	96
3.5	Estimated Emission Matrix	97
4.1	Basic protocol of the simulation	112
4.2	Density of different flows	117

LIST OF TABLES

List of Algorithms

1	Procedure of traditional gradient descent learning	38
2	Procedure of stochastic gradient descent learning	38
3	Procedure of Mini-batch gradient descent learning	39
4	Insert action for a veh-graph	46
5	Delete action for a veh-graph	47
6	Adaptation Method in 1-dimension	70
7	Adaptation Method in 2-dimension	75

LIST OF ALGORITHMS

Abbreviations

FFNN - Feed Forward Neural Network

HMM - Hidden Markov Model

ITS - Intelligent Transportation Systems

LSTM - Long Short-term Memory

MM - Mobility Model

NN - Neural Network

OD - Origins and Destinations

RNN - Recurrent Neural Network

RWP - Random Waypoint Model

UML - Unified Modeling Language

VANETs - Vehicular Ad hoc Networks

VMM - Vehicular Mobility Model

V2C - Vehicle-to-Cloud

V2I - Vehicle-to-Infrastructure

V2P - Vehicle-to-Pedestrian

V2V - Vehicle-to-Vehicle

V2X - Vehicle-to-Everything

ABBREVIATIONS

General Introduction

Scope of the thesis

This thesis is dedicated to providing a more realistic and a "better" mobility model in an intelligent transportation system and to provide an approach to bridge up its micro and macro impacts. This "better" will be specifically discussed in detail in the following of this thesis.

According to the ancient Chinese literature and records, the word traffic comes from the crossing of roads and movement of human. Since then, people moves from here to there, by foot, by horse, by wagon, etc. After the industrial revolution, vehicle became the major participants of traffic and thus the prototype of modern traffic appeared. Researches on relevant areas started at the same time.

Despite the long history of the study of traffic or say transportation systems, the fundamental purposes have never changed too much. With the continuous expansion and induction, it can be expressed as the three aspects: safety, fast and environmental friendly.

To address these major purposes and to solve the corresponding problems, such as to reduce the occurrence of traffic accidents, to ease the impact of traffic congestions and to decrease the emission of traffic pollution, the concept of intelligent transportation systems (ITS) proposed by the US Department of transportation, comes into being. Several similar traffic systems have also been proposed after, and the researches upon it have gain lots of interests recent years.

ITS contains a very wide research interests and plenty of different research fields and subjects have been involved, which makes ITS an even more complex system so that it could be regarded as an independent discipline.

GENERAL INTRODUCTION

Specifically speaking, ITS covers both microscopic and macroscopic level. One of these ideas is to make full use of the current hardware in a transportation system and to equip sensors and cameras to acquire information for decision making and then to provide assistance to the human driver or to fulfill complete automatic driving behavior on road. There are also other ideas, for example the non-driver thinking, which is based on the assumption of future evolution of the hardware of a transportation system. The existing traffic lights, traffic signs, etc. are designed for human drivers and these traffic infrastructures limit the performance of an automatic driving, such as the eyesight range depends on the weather conditions. In the V2X assumptions, the vehicles and the infrastructures can communicate with each other in another high efficient method, then comprehensive information for automatic driving could be obtained, meanwhile an optimization arrangement of the vehicles can also become possible. As for a macroscopic perspective, there is also existing applications of ITS, such as the VehiLux project, which uses open-world map data and generates macroscopic vehicle traces. These traces are particularly useful in a Vehicular ad hoc network (VANET) simulation.

Great deal of examples remains, but we would rather set them aside now. Researches presented in this thesis focus on the mobility model, which is traditionally used in a microscopic level traffic simulation and this model describes the behavior pattern of vehicles. The traditional mobility models have already been studied, which is normally an optimization model based on the three major purposes: safety, fast and environmentally friendly. However, considering possible applications at current stage, the realistic level could be another measurement of the model. Combining with machine learning methods, a more realistic mobility model could be obtained. Since the mobility model is the critical point to address the applications including but not limited to trajectory prediction, behavior simulation, meanwhile prediction of trajectory remains essential useful in automatic driving and assistant driving. As long as the V2X assumption is not completely located, to discover a mobility model at a convincing realistic level is important and necessary for the virtual scenario traffic simulation.

Researches in this thesis locate at the scope of microscopic mobility model in intelligent transportation systems, combining with the use of machine learn-

ing methods in computer science and technology to extract and depict the realistic level of the proposed mobility model. Besides, the validation of the mobility model in a virtual scenario traffic simulation needs the support of certain relevant simulation techniques, including but not limited to the animation presentation platform, and details that should be considered in practical implementation. Combining all these, the mobility model allows bridging up the microscopic level behavior and macroscopic level planning.

Main contributions

The main contributions of the thesis are summarized as follows:

- A data-driven mobility model is proposed to characterize vehicular behaviors in extreme situations. The basic thinking of this method is not limited to extreme situations such as bridges, tunnels or highways entrances (exits). On the contrary, the proposed model shows obvious advantages in these situations, which bring unknown and unconsidered issues that may cause significant error if using traditional models.
- Probabilistic represented mobility models are under consideration in this thesis. Traditional mobility models intent to show the vehicular behavior through the use of partial differential equations (PDE), which meet the dynamics and kinematics constraints of corresponding situations and certain hypothesis. An adaptation method of these models to the probabilistic represented models are given in this thesis, so that it is possible to observe and review the probabilistic distribution of possible performances, since the probabilistic represented models are more closely coupled with machine learning method.
- With the assumption that the driving state can be described by Markov property, Hidden Markov Model (HMM) is introduced into the proposed data-driven mobility model to improve its performance. The probabilistic mobility models are also used to represent the vehicular behaviors in an HMM.

- A simulation platform based on the Unified modeling language (UML) is designed and used. Both the static structure and the dynamic process of this platform can be found in corresponding UML diagrams. Besides, vehicles are organized according to a proposed algorithm and structure during the simulation. Training of neural network is carried out in an open-source tensorflow project by Google while the Simulator SUMO (Simulation of Urban Mobility) is also used to present the simulation results. Matlab was chosen as the major simulation programming language to connect and communicate through the different parts of the platform.

Outline of the thesis

This thesis is divided into 5 chapters:

In **Chapter 1**, brief introduction of the intelligent transportation systems (ITS) would be given, including the development of ITS and various motivations of previous work. Intelligence has been added into the transportation systems based on different purposes, which are safety, efficiency and environmental friendly. Applications on these purposes will be reviewed. Meanwhile, the motivation of this thesis will also be discussed.

In **Chapter 2**, a neural network based data-driven mobility model is presented. This mobility model itself is simple, but it is the fundamental of the following method. Objective of this mobility model is to extract the historical driving pattern from the existing traffic trajectory data. An application of neural network will do the work, and the applications of neural network on ITS will also be reviewed.

In **Chapter 3**, an HMM is introduced to the previous proposed model. In order to address this integration, an adaptation of traditional mobility model is necessary. Objective of this adaptation is to keep the model meanwhile change the form. Context of traditional mobility models is also presented. Comparing with traditional models, their probabilistic adaptation present the same model, however its probabilistic description will show advantages when being used in a machine learning context. Basic knowledge of HMM and its applications in ITS is reviewed in this chapter. Knowledge on the adapted traditional model,

which is regarded as a definite state of HMM, will make contribution on the initialization of the HMM so that its estimation will be more accurate.

In **Chapter 4**, a simulation scenario based on algorithms from previous chapters is described. The design of our simulation environment will be given by UML: both static and dynamic properties. Some mechanisms in practice will also be discussed and the traffic scenario simulation using our mobility model will be derived, including a highway scenario from database to evaluate the realism of our model, an intersection scenario to demonstrate the application of our model in a mixed model and a lane merge scenario to show the application of our model in transportation prediction.

GENERAL INTRODUCTION

Chapter 1

Introduction

Contents

1.1 Background	7
1.1.1 Motivation	7
1.1.2 Intelligent Transportation Systems	8
1.2 Mobility Models in Vehicular Ad hoc Network	13
1.2.1 Brief introduction to VANET	13
1.2.2 Major concerns of VANET	14
1.2.3 Self-Driving Cars and ACC systems	16
1.3 Challenges, Opportunities and Connections	19
1.3.1 Realistic level: another starting point	19
1.3.2 Traffic prediction with Deep Learning approaches	21
1.3.3 Relationship with Self Driving	22
1.4 Conclusion	22

1.1 Background

1.1.1 Motivation

Transportation has been known as a research subject and been well studied for a long time. Putting aside the academic purposes, transportation plays such an important role in human societies.

1. INTRODUCTION

As a result of modernization and urbanization, transportation system grows and becomes bigger, heavier and much more sophisticated. Maintaining and organizing of transportation system bring challenges indeed, and sometimes it could still turn out to be useless operations, which will result in inefficient traffic. Some of these inefficiencies would be ascribed to the incomplete consideration, while the others are the pure responsibility of the urbanization. Vehicles per capita, as shown in Figure 1.1, have reached an insane number now. According to the statistical research, the road motor vehicles per thousand inhabitants reaches 910 in United States in 2017, and over 500 in most of the European countries. As for developing countries such as China, this number in 2016 would be almost five times larger than it was ten years ago (in year 2006). More and more data and information indicate that a wave of ripples in the transportation may eventually affect the global economy and have far-reaching effects.

Due to the importance mentioned above, people have been conducting researches on transportation and transportation systems for a long time. Although these researches have a long history, this domain regains many attractions recently especially because of the breakthroughs in computer science. Improvements on both computer hardware and software bring out more capabilities in calculation. As a result, the concept of intelligent transportation, which takes the transportation into a higher level, is proposed. Using the deep learning technology, intelligent transportation is one step closer to its realization.

1.1.2 Intelligent Transportation Systems

People have their demands for intelligent transportation in various ways. As a fruit of such demands, intelligent transportation systems (ITS) is presented as an answer. In summary, we could conclude that a transportation system with a safety guaranteed property, an efficient mechanism in dealing with congestion problems and a solution to environmentally friendly emission issues is in need. Facing all these issues of current road transportation systems, ameliorations in both system level and individual level need to be investigated.

ITS has gained a huge amount of attention in recent years. It can be broadly defined as the application of advanced technologies to supply real-time infor-

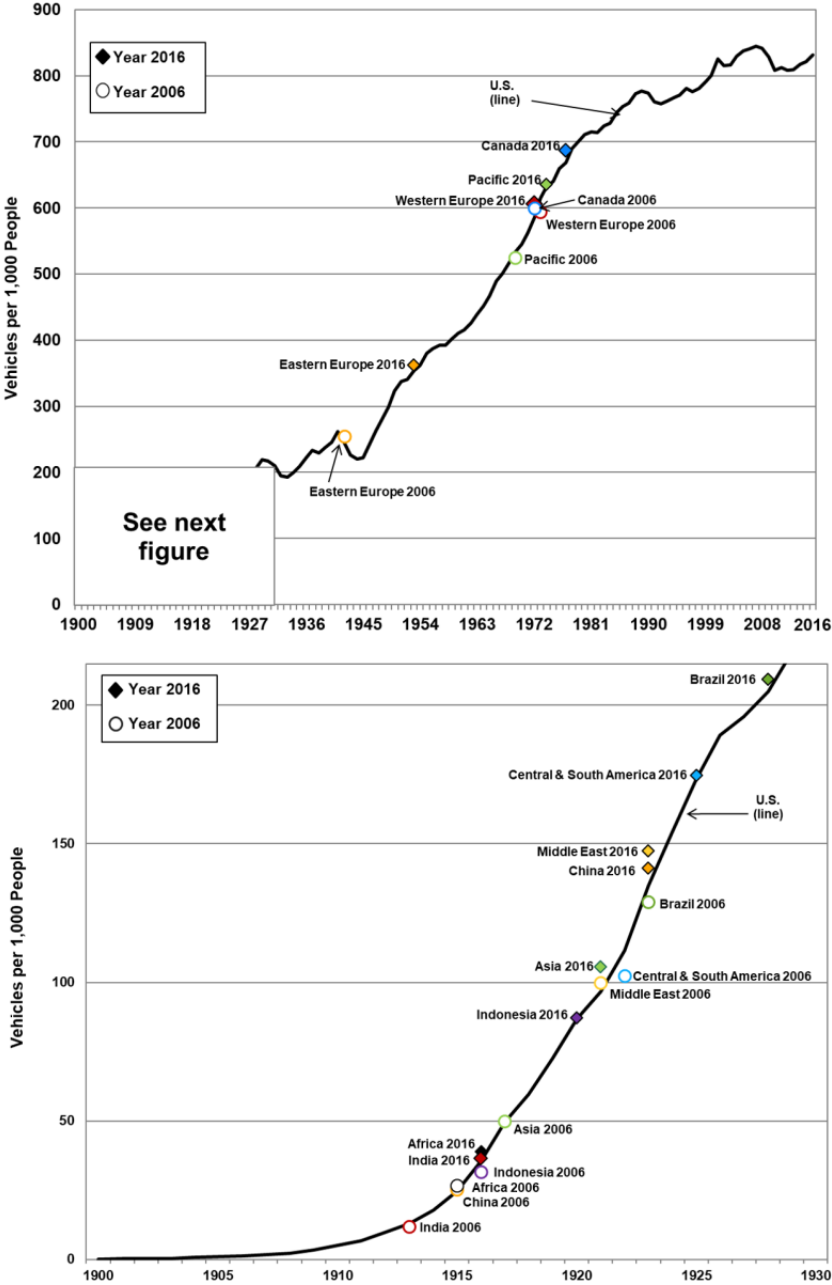


Figure 1.1: Vehicles per thousand people: US (over time) and other countries (in 2006 and 2016)
 (Source <http://www.greencarcongress.com/2018/10/20181010-fotw.html>)

1. INTRODUCTION

mation, to improve safety, efficiency and comforts, at the same time reduce its impact to environment (Perng *et al.* (2011)). It is an interdisciplinary study. Benefit from its wide coverage, information technology, communication technology, sensors, control, and computer science, etc. have made a contribution to the research of ITS. This interdisciplinary nature increases the problem's complexity due to the requirement of knowledge transfer and cooperation among different research domains (Figueiredo *et al.* (2001)).

As a consequence, ITS contains a wide range of research interests. It can be found among previous work on ITS that there is a diversity kind of problems that have been faced. For example, the routing or communication problems of the network focus on the network or its communication protocols for either vehicle-to-vehicle communication or vehicle-to-infrastructure communication (Li & Wang (2007)). Game theory is also employed in dealing with the traffic light management and organization problems (Zhao *et al.* (2012)) in order to ease traffic congestion. Besides, Liu & El Kamel (2016) provide a platoon control method to adjust the speed of vehicles so as to fulfill the blind crossing of an intersection. It is also possible to find research on the interests such as length estimation problem of the waiting platoon (Liu *et al.* (2009); Tiaprasert *et al.* (2015)), co-operative traffic monitoring problem, etc.

In short, all these research on ITS are committed to either improve road safety, relieve traffic congestion or comfort driving experience (Baskar *et al.* (2011)). Safety, efficiency, environment friendly can be summarized as the three basic elements of ITS. Each of them comes from a real-world practical problem, and leads to an issue that deeply concerns the performance of a transportation system.

When talking about the traffic safety issues, traffic accidents take away a considerable number of human lives every year and even more injuries. These accidents directly cause a huge quantity of families to be incomplete and bring deep sorrow to their relatives which deals mental damages unseen. To calm the relatives and to recover the accidents shall cost quite a fortune, yet not to mention the immeasurable impact on the related economical loss, which could be linked to causing congestion and inefficiency on the implicated logistics system. Increasing traffic has produced a growing number of accidents and fatalities. Nearly 1.3 million people die in road crashes each year (WHO (2018)),

on average 3,287 deaths a day, and 20-50 million are injured or disabled. The majority of accidents are caused by incorrect driving behaviors, such as violate regulations, speeding, fatigued driving, and drunken driving.

In another aspect, lots of examples on traffic efficiency exist. Such as the EURO CUP 2016 held in France, the event itself brought huge traffic impact to the local traffic. Settling such a heavy traffic system is not just considering about the time and congestions, in fact, the space consuming is also another subject that means. In order to deal with congestion and parking difficulties due to the increasing traffic loads, streets are widened and parking areas are built, which seizes the space for public activities like markets, parades, and community interactions.

One step further, when congestion appears, vehicles move at a low velocity. Considering the green traffic concept proposed in recent years, the congestion may bring much more emissions than a smooth passage road so that it cost more energy than it does in a smooth passage. Additional energy consumption is the one last thing that modern society would like to see. Air pollution and noise pollution are the by-products of road transportation systems, especially in metropolis where vehicles are considerably gathered. Smog brought by vehicles, industries and heating facilities is hurting people's health. The exhaust from incomplete combustion when the vehicle is in congestion is even more pollutant.

To address these aspect, meta-type of ITS has been proposed. As for an instance of ITS, project "VehiLux" ([Grzybek et al. \(2012\)](#)) is a very good example. The VehiLux makes full use of its sensors deployed all over the highway in Luxembourg as shown in [Figure 1.2](#). It can also generate random restoration of virtual trajectories. The system contains both a microscopic level using mobility models and a macroscopic level. At the macroscopic level, traffic demand is defined in terms of origins and destinations (OD) of traffic flows. Traffic mobility is time-variant and presents different characteristics dependent on a considered scenario.

Another instance of an architecture of road ITS for commercial vehicles is designed as in ([Intel \(2015\)](#)). This system is used to reduce fuel consumption through fuel-saving advice, maintain driver and vehicle safety with remote vehicle diagnostics and enable drivers to access information more conveniently.

1. INTRODUCTION

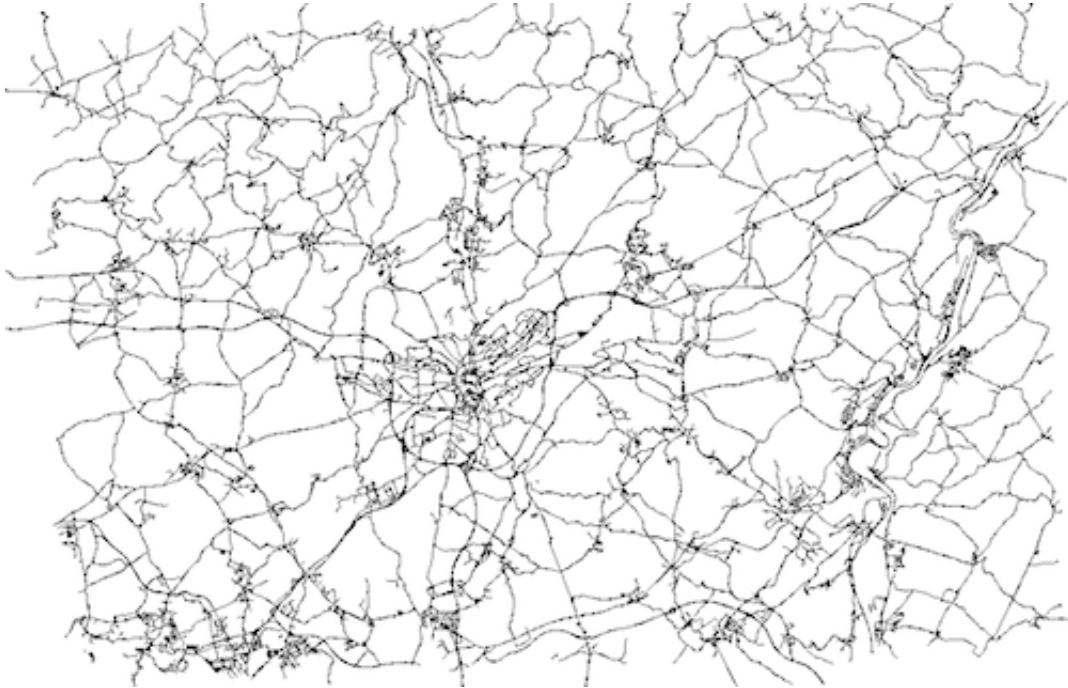


Figure 1.2: Sensors planted in Luxembourg area
(Source <https://vehilux.gforge.uni.lu/index.html>)

As shown in Figure 1.3, there are three layers in this system:

- *Sensing Layer*: this layer employs a vehicle terminal which enables the interaction between the driver and the vehicle. Meanwhile, it acts as a gateway for in-vehicle technologies and sensors, including microwave detection, speed sensor, radio frequency identification, camera, monitoring equipment, etc.
- *Communication Layer*: this layer ensures real-time, secure and reliable transmission from a vehicle to the service layer via different networks, such as 3G/4G, Wi-fi, Bluetooth, wired networks and optical fiber.
- *Service Layer*: in this layer, diverse applications using various technologies are implemented, such as cloud computing, data analytics, information processing and artificial intelligence. Vehicle services are supported by a cloud-based, back-end platform that has a network connection to vehicles and runs advanced data analytic applications. Different categories

1.2 Mobility Models in Vehicular Ad hoc Network

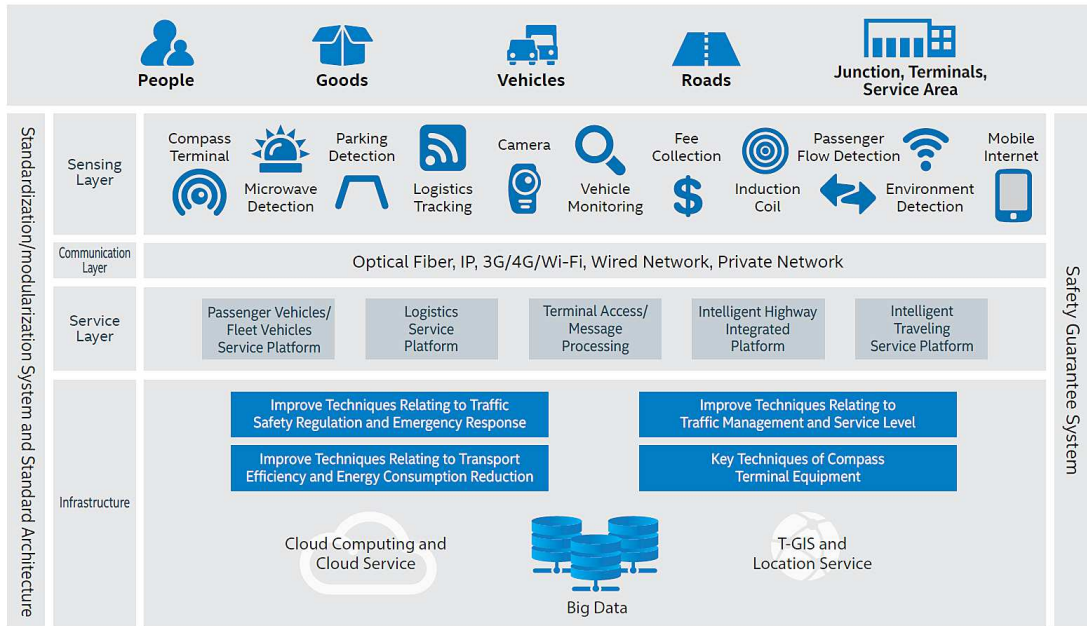


Figure 1.3: Instance for road ITS system layout

of services can be supplied, including collision notification, roadside rescue, remote diagnostic, positioning monitoring.

1.2 Mobility Models in Vehicular Ad hoc Network

In an ITS, attention is paid to both individual intelligence and systematic intelligence because of the different research focuses. Individual intelligence refers to the intelligent vehicles, the self-driving car and cruise aid system, while the systematic intelligence means the implementation of vehicular networks.

1.2.1 Brief introduction to VANET

Although we are so eager to achieve system-level intelligent transportation, it cannot be accomplished by simply replacing the vehicles on road with intelligent vehicles. One of the key features in ITS is the communication ability of all the participants which enables information exchange among them. A Vehicular Ad hoc Network (VANET) is a part of the whole ITS framework. It is an

1. INTRODUCTION

ideal network model to represent the communication among vehicles, while vehicles are regarded as the mobile nodes within the network.

As for a real-world vehicle, driver makes his decision according to his judgment, which is based on his acknowledgment of his surrounding. There are methods that dedicate to path planing meanwhile concern the modeling of sight range (Geraerts & Schager (2010)). This kind of vehicle can be represented by an intelligent vehicle, and the sight range depends on the performance of its sensors. However, suppose the driver is aware of the whole situation, it is possible for him to make the global optimal decision, which will greatly increase the efficiency of the whole traffic system. This total awareness is possible for an intelligent vehicle if its sensors accuracy and range performance can be ensured fully functional. Assuming that, the drivers can share their sight and acknowledgment of the environment, it could help them to predict the future road condition and make even better decisions. This information share is normally represented by the communication among intelligent vehicles. Thus the individual intelligence can be turned into group intelligence and systematic intelligence.

To achieve these features, there are several kinds of such communication in VANET. The communication between a vehicle and another vehicle or non-vehicle agent is usually noted as V2X, including V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure), V2P (Vehicle-to-Pedestrian) and V2C (Vehicle-to-Cloud) (Park & Min (2015)). With VANET, we can upgrade autonomous systems to cooperative systems.

1.2.2 Major concerns of VANET

While being promising in saving time and saving lives and being conceptually straightforward, design and deployment of VANET is technically and economically challenging (Hartenstein & Laberteaux (2010)). In technical terms, its concerns include the following issues:

- High mobility of the nodes. Due to high relative speed between cars, network's topology changes very fast, and many paths disconnect before they can be used (Yousefi *et al.* (2006)). Wang (2004a,b) tried to find the

1.2 Mobility Models in Vehicular Ad hoc Network

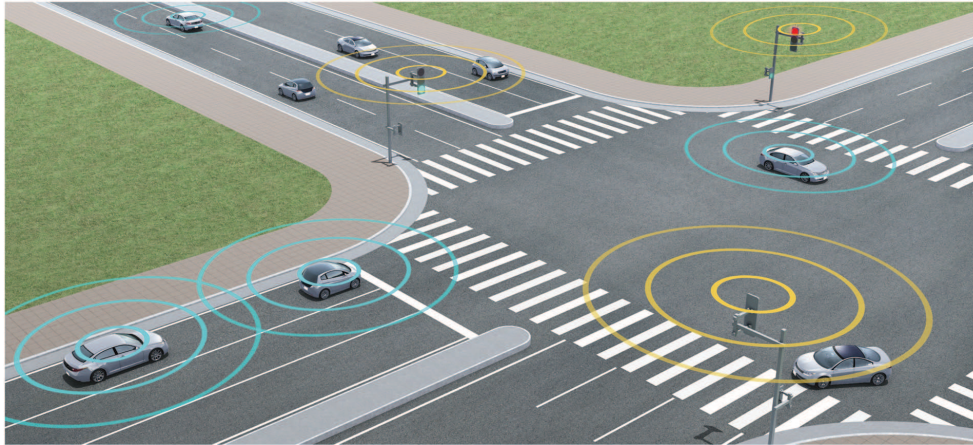


Figure 1.4: Vehicular Ad hoc Networks
(Source johndayautomotiveelectronics.com)

approximation of link's lifetime. In ([Artimy et al. \(2004, 2005\)](#)), the authors attempted to capture some relationship between the model of vehicular mobility and connectivity of the networks. In ([Blum et al. \(2004\)](#)), the effective network diameter in a typical VANET is studied.

- Self-organized structure. Lacking of centralized management and coordination entity. It is difficult to synchronize and manage the transmission events of different nodes.
- The signal's physical transmission. The transmission could be blocked by obstacles, e.g. buildings in cities, the propagation models of the signal and the influence on the performance of VANET need to be studied.
- Standardization and flexibility. Standardization is necessary to make the equipment from different manufactures compatible. while the manufactures will want to create some product differentiation.

These issues can be concluded into two major concerns. First, how to guarantee an effective communication is the number one priority of VANET. The problem in the communication part includes the difficulty in its physical implementation, and the communication protocol's design. In order to thoroughly and systematically study a VANET protocol, it is important to simu-

1. INTRODUCTION

late this protocol and evaluate its protocol performance. Thus, it leads to the second major concern of VANET, which is the mobility model (Bai & Helmy (2004)). Mobility model is associated with the protocol, it can be used to examine the impact of high mobility on the performance of VANET protocol.

A mobility model, which is only used to represent the mobility of mobile nodes in VANET in the beginning, is used to describe the behavior, and movement pattern of vehicles now. In our research, we dedicate to providing a vehicular mobility model with more realistic attribute. Such a model can make a difference in mimicking the movements of real vehicles during simulation and may play a vital and essential role in such as self-driving vehicles, trajectory prediction, and many other applications.

1.2.3 Self-Driving Cars and ACC systems

People are always interested in self-driving cars, which started long before the concept of ITS was proposed. A self-driving car is built according to the three basic elements as mentioned above: improved road safety, relieved traffic congestion and comforted driver experience (Sukthankar *et al.* (1998)). Taking advantage of the development of computer vision, online recognition and classification of objects in surrounding can be done efficiently (Levinson *et al.* (2011); Peña-Cabrera *et al.* (2005)). Then, the car can sense and perceive the surroundings using its sensors and strives to achieve more efficient vehicle operation either by assisting the driver (via advisories or warnings) or by taking complete control of vehicle (Baskar *et al.* (2011)).

Since 2003, Defense Advanced Research Projects Agency (DARPA) of USA has found a prize competition "Grand Challenge" to encourage the development of technologies needed to create the first fully autonomous ground vehicles. The Challenge required robotic vehicles to travel a 142-mile long course through the desert within 10 hours. Unfortunately, in the first competition, none of the 15 participants have ever completed more than 5% of the entire course. while in the second competition in 2005, five of 23 vehicles successfully finished the course, and "Stanley" of Stanford (see Figure 1.5) became the winner with a result of 6 h 53 min (Russell (2006); Thrun *et al.* (2006)). This robotic car was a milestone in the research for modern self-driving cars. Then it comes to the "DARPA Urban Challenge" in 2007. This time the autonomous

1.2 Mobility Models in Vehicular Ad hoc Network



Figure 1.5: Stanley at Grand Challenge 2005

vehicles should travel 97km through a mock urban environment in less than 6 hours, interacting with other moving vehicles and obstacles and obeying all traffic regulations (Montemerlo *et al.* (2008); Urmson *et al.* (2008)). These vehicles were regarded as the initial prototype of Google self-driving cars.

In 2010, European Research Council sponsored a project: VisLab Intercontinental Autonomous Challenge (VIAC) to build four driver-less vans to accomplish a journey of 13,000 km from Italy to China. The vans have experienced all kinds of road conditions from high-rise urban jungle to broad expanses of Siberia (Broggi *et al.* (2012)).

Google also started world wide known self-driving car project. It is considered to be currently the most successful project in the domain (Guizzo (2011)). Using Google's map service, the vehicle equipped with a GPS can obtain its location. Besides, a laser is installed to generate a detailed 3D map of the environment. The car then combines the laser measurements with high-resolution maps of the world, producing different types of data models that allow it to drive itself while avoiding obstacles and respecting traffic laws. More details in terms of sensors can be seen in Figure 1.6. It should be noticed that Google's approach relies on very detailed maps of the roads and terrain to determine accurately where the car is. When GPS information got disturbed, awful mistakes could happen.

Similarly, in China, the company Baidu announced its autonomous car has successfully navigated a complicated route through Beijing (Davies (2015)).

1. INTRODUCTION

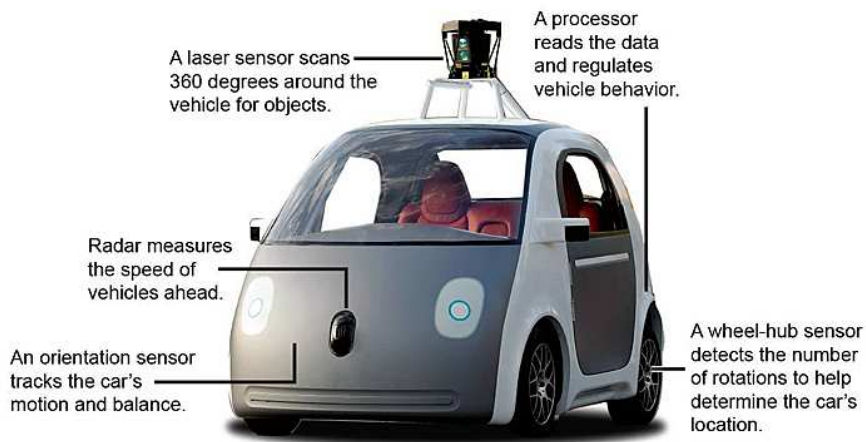


Figure 1.6: Sensors on Google's self-driving car (Source www.aventurine.com)

The car drove a 30 km route around the capital that included side streets as well as highways. The car successfully made turning, lane changing, overtaking, merging onto and off the highway.

The commercialization of self-driving cars cannot be done without automobile manufacturers. Some of them have launched their own self-driving projects targeting different scenarios (Chandler (2016)), such as "Drive Me" of Volvo (Ziegler (2015)), "Buddy" of Audi (Davies (2014)), Tesla (Korosec (2016)), etc. These prototypes are still at test stage, but it is a necessary step of self-driving car development.

Along with self-driving cars, a set of advanced driver-assistant systems (ADAS) are developed, include the very important adaptive cruise control (ACC) system (Vahidi & Eskandarian (2003)). ACC is now commercially available in a wide range of passenger vehicles.

ACC systems are an extension of cruise control (CC) systems. CC is used to maintain vehicle's velocity to a decided value, and the driver does not have to depress the pedals, therefore the driver can be more focused on steering wheel. CC can be turned off both explicitly and automatically when the driver depresses the brake. As for ACC, if there is no preceding vehicle within a certain range, it works as a CC system; else, it uses the range sensor to measure the distance and the relative velocity to the preceding vehicle. Then after calculation, if the preceding vehicle is too close or is traveling slowly, ACC

1.3 Challenges, Opportunities and Connections

shifts from velocity control to time headway control by control both the throttle and brake (Xiao & Gao (2010)). Of course, ACC still has its own limits: in general, ACC system is limited to be operated within a velocity range from 40km/h to 160km/h and under a maximum braking deceleration of $0.5g$ (Rajamani (2011)). The operations outside these limits are still in the charge of driver, because it is very difficult to anticipate the preceding vehicle's motion only by using range sensors. Yet with the help of V2X communication in a VANET model, the distance acquirement can be more precise, and the systems turns into a cooperative ACC (CACC) system De Bruin *et al.* (2004); Lu *et al.* (2002); Xu & Sengupta (2003).

Despite the differences between completely self-driving and assistance driving, self-driving vehicles and ADAS with the example of ACC-series systems share the same purpose and use a similar methodology. Comparing these with mobility models, mobility model locates itself into a more abstract level: it focuses on the attributes presented such as velocity. While ACC and self-driving vehicles cannot get away with the dynamics model of vehicle, it locates in a more actual level.

1.3 Challenges, Opportunities and Connections

In this section, we will discuss the major challenges and the possible opportunities of realizing a realism based mobility model, and the connections between such a mobility model and its potential application in self-driving.

1.3.1 Realistic level: another starting point

To begin with, the three major elements of ITS shall be reminded: safety, efficiency, and environmental friendly. These three aspects cannot be seen as completely disjoint sets of features (Hartenstein & Laberteaux (2010)), they can transform to each other under certain conditions, and they can explain each other in certain scenarios. For example, vehicle crashes will lead to a traffic jam. A message reporting an accident can be regarded as a safety message for the near-by vehicles on one hand and can also be regarded as an input to find an alternative route in a transport efficiency application on the other

1. INTRODUCTION

hand. Traffic inefficiency will result in extra resources consuming, which will lead to more emissions. Bad traffic organization leads to bad road condition. As a result, frequently speed changes will happen, which leads to not only congestions but also extra safety risks since the collision chances increases.

Major existing works that are rooted in either of these three aspects, could be seen as that they are from the same purpose. For example, [Liu & El Kamel \(2016\)](#)'s work aims at the blind traverse of an intersection. Its directly target would be the safety, meanwhile the efficiency of intersection traverse increases. [De Schutter & De Moor \(1998\)](#) presents a traffic light control method for a two by two way street intersection. Schedule of the traffic light is calculated to optimize the average queue length. It focuses on the efficiency, meanwhile the performance of safety at this intersection augments indirectly.

In short, the existing works are more or less dedicated to optimizing on one of these three aspects. However, it could be seen from a totally another point of view. The realistic level of the output behaviors has not been mentioned a lot, it can also be used as the measurement of a model. It is important to use a realistic mobility model so that results from the simulation correctly reflect the real-world performance of a VANET ([Karnadi *et al.* \(2007\)](#)). In [Härri *et al.* \(2006\)](#)'s work, a VanetMobiSim is proposed as a generator of realistic vehicular movement traces for telecommunication networks simulators. Real-world database TIGER (Topologically Integrated Geographic Encoding and Referencing) ([Bureau \(2018\)](#)) is used to enhance its previous model CanuMobiSim in representing macro-mobility features. Similar things have also been done in ([Saha & Johnson \(2004\)](#)), where TIGER is also been used to provide geographical information and special attention has been paid to the characteristics of a realistic street model, and in ([Jardosh *et al.* \(2003\)](#)), where Voronoi diagram is used to determine pathways through obstacles, thus this model produces better traces than standard random waypoint model.

The NGSIM (Next Generation of Simulation) database is also widely used in realism based model. Differs from the TIGER database, the NGSIM database provides data of vehicles behavior in microscopic level. However, the NGSIM database does not provide vehicle trajectory data directly, the dataset is built for vehicle tracking. Background segmentation ([Osorio *et al.* \(2013\)](#)) and object extraction methods ([Zhang & Lenders \(2002\)](#)) are necessary to recognize

1.3 Challenges, Opportunities and Connections

the vehicles in the surveillance image data. Besides, it is also possible to use near-accurate reconstructed data from (Montanino & Punzo (2015); Punzo *et al.* (2011)), which specifically focus on NGSIM data retrieve.

To mimic the real world behavior, if the real world situation has been well observed, then it will not be hard to find out that the behaviors taken are the good solutions instead of the optimal solutions. Thus, using the optimal models are not very proper. In order to evaluate the realistic level of entity behaviors, a data-driven based method is commonly used (Charalambous & Chrysanthou (2016); Wolinski (2016)). However, since this kind of method depends very much on the selected dataset, there is no universal evaluation method. If real-world data exist, then the realistic level can be measured by a comparison between the model and data. If there is no existing data, which is the common situation, then the virtual traffic flow generated by the model is examined that if it can meet the property described by traffic flow theory in macroscopic level. Our proposed models in this thesis are based on a realistic target. The comparison with corresponding real-world data and the examination on traffic flow theory of our model will be given in the following chapters.

1.3.2 Traffic prediction with Deep Learning approaches

Another important application of the realistic model is in traffic prediction. A good traffic prediction has the potential to improve traffic conditions and reduce travel delays by facilitating better utilization of available capacity. Since deep learning method with neural network model will be used in the following chapters to produce our realistic convincing mobility models, some representative uses of neural network in traffic prediction methods will be examined in this section.

Song *et al.* (2016) constructed a deep recurrent neural network to predict the human mobility. Satellite GPS records are collected and used, which means the human mobility in this model may contain mobility with various kinds of transportation, such as by foot, by car or by train.

Lv *et al.* (2015) however, makes the prediction for traffic flows. Stacked autoencoders structure is used to learn the patterns in the traffic data with training performed by a greedy layer-wise fashion. Likewise, Ma *et al.* (2015)

1. INTRODUCTION

uses a long short-term memory neural network for traffic speed prediction. This prediction is built based on the remote microwave sensor data.

Although these methods and our method in this thesis show some degrees of similarity, the traffic predictions normally come with traffic flow, as a consequence, it will be always acted on macroscopic level, such as in (Lv *et al.* (2015)). There is no individual trajectories there. In other representative examples, such as (Song *et al.* (2016)), GPS records from satellite can only produce rough trajectories data. As for the result, it is more likely to the Original-Destination structure than a detailed trajectory.

1.3.3 Relationship with Self Driving

As claimed in VANET, if global acknowledgment of the whole environment is known, then it will be more flexible to plan the future paths. However, this requirement somehow stands in ideal model. In path planning of self-driving, which is a more practical application, the behavior of surrounding vehicles are usually marked as unknown.

In the domain of self-driving, a realistic mobility model in microscopic level is highly demanded. Currently, static position of the adjacent vehicle is not hard to acquire, but their future dynamic trajectory is not considered predictable. Practically, it will be regarded as simple car-following models (Xu *et al.* (2017)). Deep learning is also widely used in self-driving (Houenou *et al.* (2013)), but the difference is that it is used mostly in order to complete a pattern recognition task, instead of prediction.

1.4 Conclusion

This chapter gives an introduction to the scope of our work. Firstly, the background information is introduced, that our research combines the research in transportation and deep learning. Due to interdisciplinary, the ITS plays an important role in the research of transportation. Normally, the core purposes of ITS consist of three main aspects: safety, efficiency and environmental friendly.

Secondly, we have introduced the VANET as a part of the whole ITS framework. There are two important points in VANET model: protocols and mo-

bility models. The former focuses on the communication model and its implementation, while the latter describes the movement of a mobile node in abstract (or a vehicle in practice). It is also the major concern of our research. By analyzing the composition and its relationship, we indicate that mobility model also has essential applications in individual intelligence, including both self-driving cars and ACC systems.

Thirdly, we talked about some related interests and inspirations of our research. To begin with, the three elements of ITS can convert to each other, yet we choose none of them for performance analysis of our research. We would like to use the realistic level to evaluate a model instead. Then, the demand of a realistic model in traffic prediction is clarified, and the difference between current representative research on traffic prediction with deep learning and behavior produced by our mobility model: major existing traffic prediction is always at a macroscopic level, while we focus on a realistic microscopic level model. At last, we claim the connection between mobility model and self-driving. In self-driving, since it is ego-centric, the behavior of adjacent vehicles is no predictable or using simple models. A realistic mobility model as we do may help self-driving to fulfill the trajectory prediction of other vehicles.

A conclusion can be made that the mobility model is the core of our research. It comes from the VANET structure, which is part of the whole ITS framework. The mobility model describes the behavior of a single vehicle. It may be associated with self-driving and ACC systems. Deep learning method will be combined since the realistic model is in demand. The detailed method to achieve such a model will be introduced in the following chapters of this thesis. As far as the model is concerned, it differs, yet still has connections with the traffic prediction, and it has essential applications in self-driving.

1. INTRODUCTION

Chapter 2

Neural network based data-driven mobility model

Contents

2.1	Introduction	25
2.1.1	Neural Network in intelligent traffic	27
2.2	Problem descriptions - Why using NN	28
2.3	Neural network structure and backpropagation algorithm	30
2.4	NN Application in traffic simulation problem	34
2.4.1	Pre-processing the obtained data	35
2.4.2	Deep Learning Method	37
2.4.3	Learning Process	40
2.4.4	Implement simulator for virtual traffic flow	43
2.4.5	Performances	45
2.5	Discussion and Conclusion	51
2.5.1	Discussion	51
2.5.2	Conclusion	52

2.1 Introduction

In this chapter, we will introduce a realistic mobility model. Its realistic property comes from the neural network learning.

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

First, we will discuss the advantage of a data-driven model with neural network. Then we explain the structure of a neural network and the algorithm to train a neural network so that we can introduce our specified application with a neural network in the next step. In the end, a demonstration application of this mobility model on virtual traffic simulation will be given.

More specifically speaking, traffic simulation can have different interpretation. Previous work of [Yu *et al.* \(2014a\)](#) has combined conventional traffic motion model with virtual reality technique, and produced a traffic simulator for driving scenarios. It can also be as the virtual traffic system that is represented as an autonomous agent system, similar to a crowd simulation system. According to ([Reynolds \(1999\)](#)), a well-rounded agent system could be built in three layers: the action selection layer, the steering layer and the locomotion layer. In this chapter, we try to carry out a novel traffic simulation system with the realistic mobility model. The simulation refers to produce virtual, reliable, and realistic vehicular flow. This kind of vehicular flow has a critical use in predicting, pre-casting and simulating a certain traffic situation for the specified transportation scenario as mentioned in previous chapter. The simulation aims to build up a bridge between the two kinds of view of daily traffic: the macroscopic traffic and the microscopic traffic. Changes in microscopic traffic could be easily shown when some macro managements are taken. Thus, it could help to ease the congestion and reduce the pollution, in alternative way. It also has an essential application in traffic design or traffic control for either daily life traffic or special events traffic.

For example, a special event can bring up a large amount of traffic load and causes changes to daily traffic. At the same time, a specified, well-rounded parking and traffic control plan would be in need, which is very much based on the in-coming and out-coming vehicular flows. However, since the event is not a normal action for city traffic, little previous traffic data can be used as reference. Organizing scenarios of large-scale real traffic can solve the problem, however, with a huge expense and can still be inefficient. Thus, the virtual vehicular flow is essential to simulate these scenarios.

In this chapter, we mainly focus on a vehicular mobility model using a neural network based method to produce more realistic virtual traffic vehicle flows. The proposed method may also consider the interactions among

vehicles, that although every vehicle makes its decision individually, it also considers the relative information from the adjacent vehicles. Neural network is used to learn the hidden spatial-temporal relationships among vehicles. The learned mobility model would help in providing the traffic simulation locally, which indicates that it is possible to synthesize and generate virtual trajectories for each single vehicle agent in a traffic system. The produced simulation would be more realistic than the one produced by conventional ones. A specified structure to organize and manage the virtual vehicles is implemented. The presentation by far is implemented by employing a traffic simulator SUMO.

2.1.1 Neural Network in intelligent traffic

Artificial neural network ([Hagan *et al.* \(1996\)](#)) is a computational approach which is based on a loosely modeling on the way how real neural network solves problems. As the elaboration of [MacKay \(2003\)](#), a neural network model could be efficiently used in applications such as clustering, estimating of statistical distributions, compression and filtering. Meanwhile it is also capable in applications such as regression analysis, prediction and modeling.

As for the existing work in relevant domain, [Yu & Chen \(1993\)](#) used a three layer simple neural network model with back-propagation algorithm to predict traffic. Due to the limitation of computational capability, the proposed network is quite simple in his modeling that the traffic quantity of a time period is chosen as the only feature focused during the prediction.

Recently, the network grows much bigger and more complex. [Jiang & Fei \(2015\)](#) proposed a two-level vehicle speed prediction system for highways based on neural network and hidden Markov model. In this system, the under simulated road has been divided into several segments. The traffic speed of certain road segments on certain time stamp is predicted by neural network with the information of neighbor segments using historical data. Differently, [Ma *et al.* \(2015\)](#) presented a long short-term memory neural network to predict speed using microwave detector data. To validate the effectiveness of the proposed method, one month traffic data with the updating frequency of two minutes from two sites in Beijing expressway were collected. Comparisons with other method such as SVM have also been made on the same dataset.

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

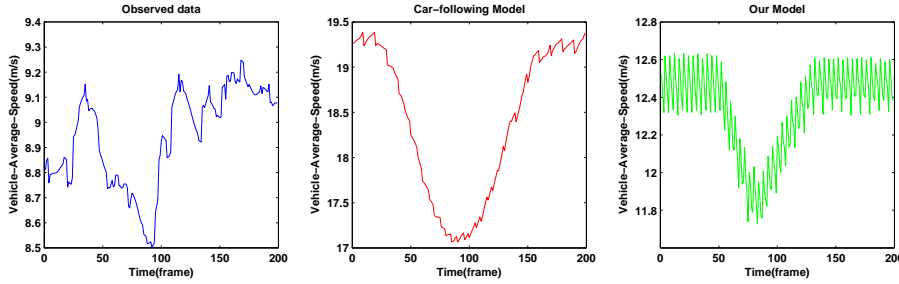


Figure 2.1: Comparison on average speed of vehicle flows with different models

Despite the diversity of using neural network in solving traffic prediction problem, there is one essential difference that all these prediction systems focus on the overall speed prediction, which is from the macroscopic-level modeling. Individual behaviors have not been considered in their methods. As for the simulation in microscopic level, [Lerner *et al.* \(2007\)](#) presented an example-based simulation technique, however for the human crowds. [Xia & El Kamel \(2016\)](#) used neural network with reverse reinforcement learning method to solve an enhanced a-star path planning problem in an unknown scenario for robot.

In our neural network usage in this chapter, we focus on the simulation and prediction in microscopic level. Differs from has been done by [Xia & El Kamel \(2016\)](#), the under-simulated target would rather be a traffic vehicular flow with multiple diverse vehicles than a single robot. It is quite different from simple duplication that the interactions between vehicles would be considered which makes the situation much more complex.

2.2 Problem descriptions - Why using NN

According to the three-layer-division of [Reynolds \(1999\)](#), people shall have corresponding solutions for each of the multiple aspects to address a complete traffic simulation system. However, without considering the other irrelevant issues, the model that describes how the vehicle behaves is the core of the research in this chapter. In order to create a virtual vehicular flow, multiple previous works have been done in the action selection layer and steering layer.

2.2 Problem descriptions - Why using NN

Yet many of them, such as the car-following model (Gipps (1981); Krauß *et al.* (1997)) that describes how single vehicle behaves while following a preceding vehicle, belong to the rule-based method.

The car-following model is a branch of vehicle control models (Brackstone & McDonald (1999); Krauß (1998)), which uses ordinary differential equations to constrain and to explain how the factors that influence the vehicles' behavior evolve throughout the whole following procedure. The factors, from the basic vehicle velocity and acceleration to psychological factors are all considered in new models. Attention should be paid that the rule-based method considers only a small set of pre-defined behaviors or situations, which means the rule-based model cannot react with those situations that are emergent or have not been acknowledged before. Since modern traffic has gain more complexities, the rule-based method loses its efficiency in facing certain traffic situations and also loses its accuracy in describing certain traffic situations. Thus, the example-based method has gained more and more interests recently along with the development of computational capability. Machine learning algorithms were applied in an example-based method to reveal the overall motion directions of the agents.

We construct an experiment to demonstrate the impact on global attributes of the traffic flow when using different local mobility models from different kinds of method: the car-following model from the rule-based method and a neural network learned mobility model from the example-based method.

The comparison is as shown in Figure 2.1. Car-following model can deal with the car following behavior ideally. However, if employing the model with large scale of vehicular flow and in a complex and unexpected situation, the result would go far from the reality.

In the scenario for this experiment, we imitated a sudden stop action happened in daily highway traffic, that an uncertain vehicle in the driving lane would suddenly hit the brake and decelerated until fully stopped. The mechanical fault would then be excluded and the vehicle would accelerate and drive ahead again. The curves in Figure 2.1 indicate the average speed of the vehicles, which are located within the surveillance area, during the simulation. Since we need to initialize the scenario by filling the scene with generated virtual traffic flow, the result from the very beginning of the simulation has not

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

been presented.

While the curve in the left is the observed data selected from real-world highway surveillance, the curve in the middle is the simulation driven by conventional car-following model, with an initialized speed at 10.0 m/s, and curve in the right is the simulation driven by our neural network learned mobility model. As shown, there are decreases when the stop happened in both three curves. The shapes of these three curves are similar, but if we focus on the value of the peak and trough of the curves, we may notice that since the real world observed data is at around 10 m/s, the data from car-following model goes up to about 18 m/s, while our model is at about 12 m/s that is closer to the observed real-world data.

We can conclude that our learned mobility model shows more realistic attributes than the car-following model of the rule-based method (Zhang & El Kamel (2018)). That is because in a rule-based method, the behavior is characterized by pre-defined rules. Any unexpected issues would have impacts in the method. While in an example-based method, it is believed that the target driving model is hidden in a black box. According to the multiple inputs and outputs, which are the real world situations and real world corresponding observations in the chapter, using a learning method could adjust the parameters and then approximate the neural network to the driving model and finally, the network would be accurate enough to represent the hidden driving model.

2.3 Neural network structure and backpropagation algorithm

A neural network is a set of neurons (as shown in Figure 2.3) organized in a multi-layer structure, as shown in Figure 2.2. "Neuron" here, inspired by the biological neuron in the brain, is the meta-computational unit in the whole structure.

Neurons have been arranged into layers. Signal (input) goes into the network by the input layer (i_1 to i_n in Figure 2.2), and comes out by the output layer (o_1 to o_n). Layers between the input and output layer are called the

2.3 Neural network structure and backpropagation algorithm

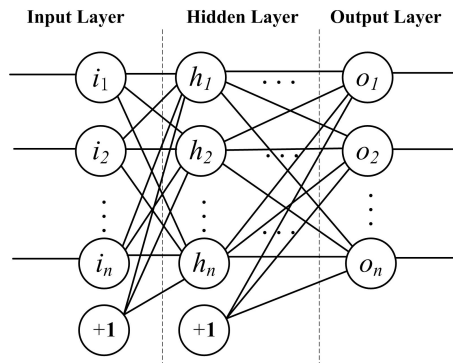


Figure 2.2: Demonstration of the architecture of Neural Network

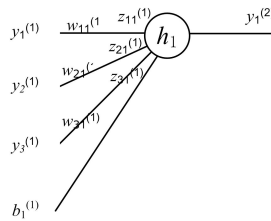


Figure 2.3: Demonstration of a single neuron

hidden layer. Neurons between different adjacent layers are usually interconnected. Links connecting neurons have a weight to indicate its strength on either increasing or decreasing the signal when it passes through this link. There is a special kind of neuron called bias, which is marked as $+1$ in Figure 2.2.

The different combinations of weights can process the input signal differently. By adjusting the weights, it is possible to carry out a certain kind of treatment on the signal, and this procedure is called the training of a neural network. According to the universal approximation theorem, a neural network with hidden layer of a finite number of neurons can be trained to approximate a random continuous function.

A bit of denotations would be necessary here. As shown in Figure 2.3, we denote the output of the i th neuron in layer l as $y_i^{(l)}$, input from the k th neuron of previous layer $l - 1$ of the same neuron as $z_{ki}^{(l)}$. Weight that connects between i th neuron in layer l and j th neuron in layer $l + 1$ is denoted as $w_{ij}^{(l)}$. Specifically, the weight from bias in layer l to i th neuron in next layer $l + 1$ is denoted as $b_i^{(l)}$.

Then, the basic relationship between these should be

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

$$y_j^{(l+1)} = f\left(\sum_i z_{ij}^{(l)} + b_j^{(l)}\right) \quad (2.1)$$

where $f(\dots)$ is the activation function and input $z_{ij}^{(l)}$ is calculated as

$$z_{ij}^{(l)} = y_i^{(l)} w_{ij}^{(l)} \quad (2.2)$$

The activation function, also called the transfer function is always non-linear and differentiable. This function maps the given the input(s) of the neuron to normally $(0, 1)$ or $(-1, 1)$. Figuratively speaking, the activation function represents the activated degree of the neuron.

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

Sigmoid function as shown in Equation (2.3) is commonly used as activation function in neural network, since its derivative respect to x has a convenient property: whose derivative can be represented by itself.

$$f'(x) = f(x)(1 - f(x)) \quad (2.4)$$

Thus, a neural network can be defined by the weights, bias and its activation function.

As for the output layer, the error E of this current network can be calculated based on its output \mathbf{o} and o_i corresponds to the i th output neuron. Suppose that error is defined as in mean square error (MSE):

$$E = \frac{1}{2} \sum_i (o_i - y_i^{(out)})^2 \quad (2.5)$$

As mentioned above, we are interested in the derivative of error E respect to the weights $w_{ij}^{(l)}$, if we want to update the weight towards its gradient direction to minimize the error.

$$w_{ij}^{(l)'} = w_{ij}^{(l)} + \eta \frac{\partial E}{\partial w_{ij}^{(l)}} \quad (2.6)$$

where η is called the learning factor or learning rate that determines the step length of updating which influence learning speed.

2.3 Neural network structure and backpropagation algorithm

In order to calculate the derivative of E respect to weight $w_{ij}^{(l)}$, by applying the chain rule, we have

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \frac{\mathbf{d}z_{ij}^{(l)}}{\mathbf{d}w_{ij}^{(l)}} \frac{\partial E}{\partial z_{ij}^{(l)}} = y_i^{(l)} \frac{\partial E}{\partial z_{ij}^{(l)}} \quad (2.7)$$

It is obvious that as in Equation (2.2), the derivative of $z_{ij}^{(l)}$ respect to its corresponding weight $w_{ij}^{(l)}$ would be $y_i^{(l)}$. Continuing apply chain rule to the obtained derivative

$$\frac{\partial E}{\partial z_{ij}^{(l)}} = \frac{\partial y_j^{(l+1)}}{\partial z_{ij}^{(l)}} \frac{\partial E}{\partial y_j^{(l+1)}} \quad (2.8)$$

where the first derivative is the derivative of activation function. If we use sigmoid function (Equation (2.3)), the derivative would be as in Equation (2.4). Then

$$\frac{\partial E}{\partial z_{ij}^{(l)}} = y_j^{(l+1)}(1 - y_j^{(l+1)}) \frac{\partial E}{\partial y_j^{(l+1)}} \quad (2.9)$$

and it can be continued by applying the chain rule to the obtained derivative

$$\frac{\partial E}{\partial y_j^{(l+1)}} = \sum_k \frac{\mathbf{d}z_{jk}^{(l+1)}}{\mathbf{d}y_j^{(l+1)}} \frac{\partial E}{\partial z_{jk}^{(l+1)}} = \sum_k w_{jk}^{(l+1)} \frac{\partial E}{\partial z_{jk}^{(l+1)}} \quad (2.10)$$

Equation (2.7) and (2.9) describe how the error E propagates through a certain neuron, and Equation (2.10) connects the error from adjacent layers. Combining all these Equation (2.7), (2.9) and (2.10), we can have a represented of derivative E respect to weights with only weights, output from neurons of passed layers and the derivative E respect to the output, according to the definition of error (as in Equation (2.5)), which would be

$$\frac{\partial E}{\partial y_i^{(l_{out})}} = \frac{\partial}{\partial y_i^{(l_{out})}} \left(\frac{1}{2} \sum_i (o_i - y_i^{(l_{out})})^2 \right) = -(o_i - y_i^{(l_{out})}) \quad (2.11)$$

Using the back-propagation algorithm (Rumelhart *et al.* (1986)) as in Equation (2.7), (2.9) and (2.10), we can achieve the derivative E respect to $w_{ij}^{(l)}$ with-

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

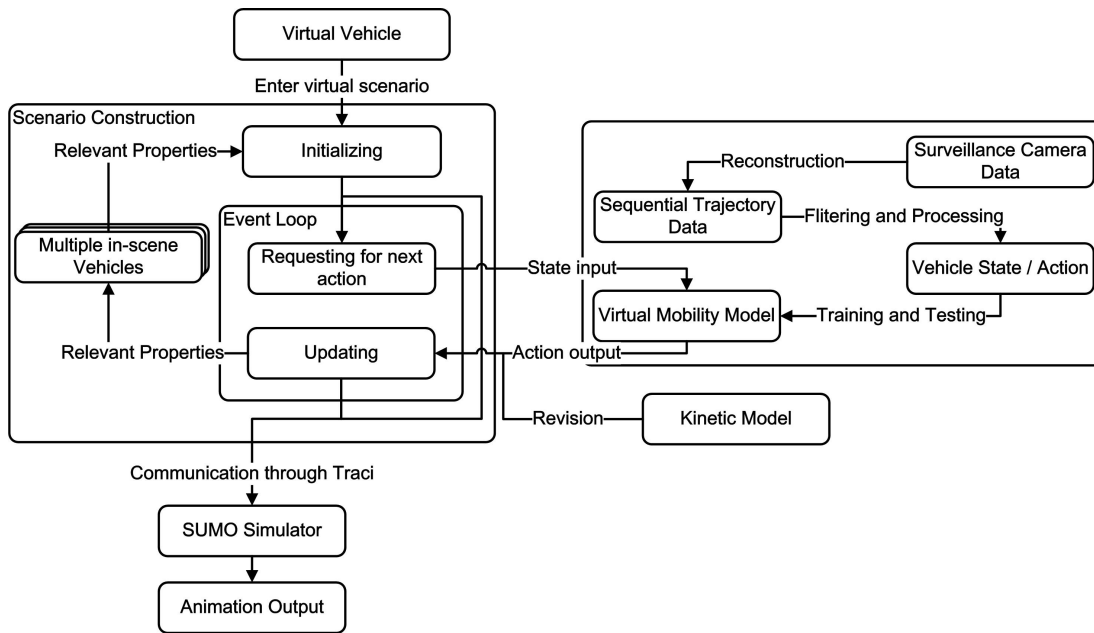


Figure 2.4: Overview of the proposed method

out redundant calculation. Thus, iterations on updating the weights to train the neural network with certain demonstration of examples could be done efficiently. It is important to realize that an initialization (Lamy & Borne (1993); Nguyen & Widrow (1990)) of the network's parameter is necessary rather than set them to all zeros. If all the weights start off at identical values, then all the hidden layer neurons will end up learning the same function of input. And the weights will be symmetrical. The initialization serves as the symmetry breaker, meanwhile, a good initialization can accelerate the learning.

2.4 NN Application in traffic simulation problem

A neural network based learning method as described in Chapter 2.3 is capable of extracting the hidden driving habits, however, there would be several parts of work should be done before and after to improve and ensure the performance of learning. For example, the observed data should be pre-processed, and how to manage and implement each part of the whole method remains a problem. In Figure 2.4, we present the overview of the system for simulation,

2.4 NN Application in traffic simulation problem

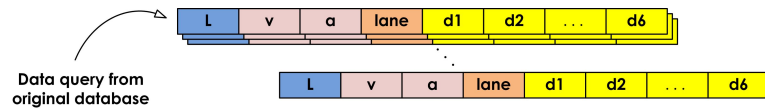


Figure 2.5: A single data query formed to represent the vehicles' state

from which the organization of each part and their relationship could be seen clearly.

2.4.1 Pre-processing the obtained data

First of all, the data query for the observed surveillance data should be selected and defined. Certain attributes would be chosen to represent the traffic in the model. Differs from using the neural network directly to the car-following model, it would be more effective to join other attributes to provide a more comprehensive traffic model. As shown in Figure 2.5, the chosen attributes include not only static properties of vehicles and its dynamics attributes, but also the relevant relationship with the adjacent vehicles. Detailed explanation of some identical variables can be found below.

1. L , the type of vehicle is various and vehicle on different type behaves differently. For example the large trucks usually don't switch its lane and continuously precede in the lane with lower speed, while a small car would rather alternate its lane more frequently. The vehicle length is a good attribute in distinguishing the different type of the vehicles.
2. v and a , the velocity and acceleration of the vehicle are essential in characterizing the driving state of the vehicle. Velocity alone is not enough to reflect the complete state of the vehicle. Both v and a could have a connection with the type of vehicle.
3. $Lane$, the current occupied lane is the discretized horizontal location data. It is more important than the local vertical distance because the interaction between vehicles may happen anywhere in the road. Besides, the local lane is the unique attribute we can achieve and observe from the original data when a lane change behavior have been performed.

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

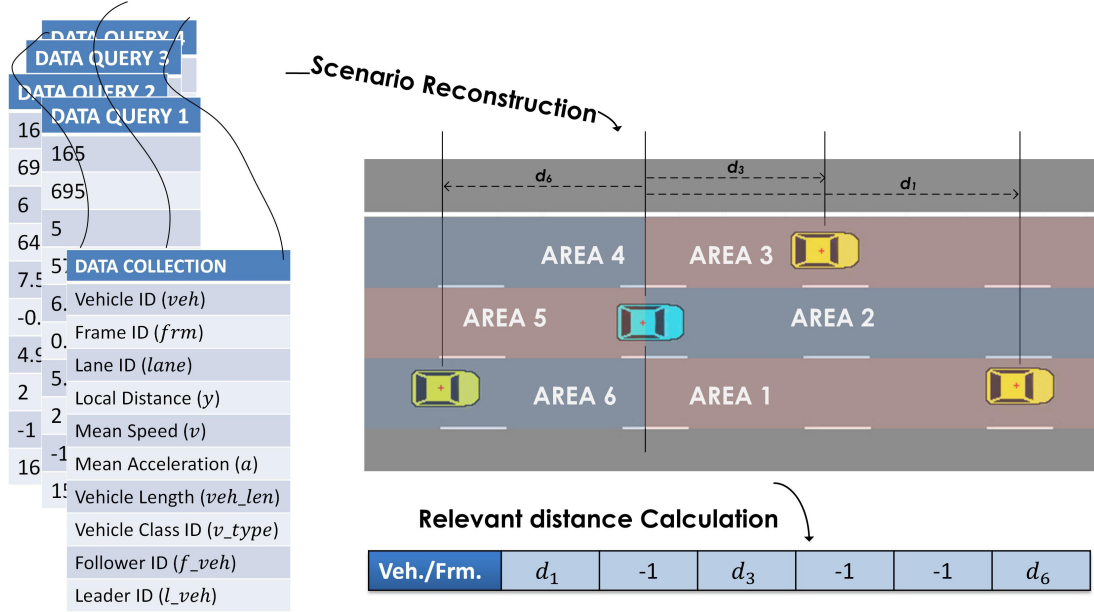


Figure 2.6: Definitions of different areas and the relevant distances

- d , the distance to the nearest vehicle demonstrates the local spatial relationship of the observed vehicle. Interactions with other vehicles or self-actions of the vehicle would have a strong dependence of this attribute. It is possible to obtain after the processing of original data.

As shown in Figure 2.6, we separate the surround areas of the observed vehicle into six: the preceding area and following area of the occupied lane and adjacent lanes. These six areas are used to define the relative position of the vehicle its surroundings. In every area, a distance to the nearest vehicle is used as the attribute to label out the situation in the corresponding area. The distance is chosen rather than the vehicle density within the area because it would be much more complex with the density. Combined with all these items, the description of a single query from the original database would be obtained as shown in Figure 2.5. The huge amount of traffic data now are normalized in representation as they are in the same structure. It could be different if other data collection would be used in the future, however, the chosen attributes should be kept since each of them is in charge of an irreplaceable category of data that could make a difference during the training.

2.4 NN Application in traffic simulation problem

The construction of data query also creates the state-action space for this simulation problem. In a high-dimensional space, a single data query stands for a current state of a vehicle. The moves from a state to another stands for the possible actions of a vehicle. These actions come from the original observed surveillance data, while they are not strict limited by the original observed data. The learned neural network would predict a proper action for a state that the input state could be new to the observed data, and also the new generated state after following the obtained corresponding action could also be new to the observed data. That's how the example-based method could deal with some complicated and unexpected situations.

A normalization of the data query is also necessary before training the network. Since different variables are chosen when forming the data query, their values float in various ranges. Unification could accelerate the training procedure of a network and also enhance the accuracy of finally obtained model. We use translation and scaling to make sure that each single attribute locates from -1 to 1.

2.4.2 Deep Learning Method

A basic neural network structure can be found as in Chapter 2.3, it is called a Feed-Forward Neural Network (FFNN). Back-propagation algorithm helps us to calculate the derivative on weights efficiently, but in order to train the neural network, a learning method will be applied to eventually decide how to update the weights. Traditional learning method can deal with the training of a FFNN, but when the network structure becomes deeper in layer and larger in scale, meanwhile the amount of data fed in to train the network increases, it is not efficient to train the network with traditional learning method. Acceleration on NN computing has been done in both software and hardware (Shkvarko *et al.* (2011)). However, after deep learning method is proposed to train a deep neural network, the neural network once again gains widespread attention. Some tricks on data processing and learning procedure are applied.

The basic concept of training a neural network is gradient descent. In practice, one update iteration of the weights will be achieved after all examples have been fed into the network. Ideally, this method can reach the global optimal point when the training lasts long enough for sufficient iterations. But

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

when the amount of data is large, the training procedure will be too long since the complexity increases. The learning procedure in Equation (2.6) will be:

Algorithm 1 Procedure of traditional gradient descent learning

- 1: Repeat until convergence: {
- 2: for each i, j and l :
- 3:

$$w_{ij}^{(l)'} = w_{ij}^{(l)} + \eta \sum_{k=1}^m \frac{\partial E_k}{\partial w_{ij}^{(l)}} \quad (2.12)$$

- 4: }
-

where $\sum_{k=1}^m$ indicates the sum of total m examples, and E_k stands for the error for the k th example.

In order to deal with this drawback of traditional gradient descent learning method, stochastic gradient descent (SGD) has been proposed ([Bottou \(2010\)](#)).

Algorithm 2 Procedure of stochastic gradient descent learning

- 1: Repeat until convergence: {
- 2: shuffle the database;
- 3: randomly choose an example k :
- 4: for each i, j and l :

$$w_{ij}^{(l)'} = w_{ij}^{(l)} + \eta \frac{\partial E_k}{\partial w_{ij}^{(l)}} \quad (2.13)$$

- 5: }
-

SGD algorithm is a drastic simplification, instead of computing the gradient exactly, each iteration estimates this gradient on the basis of a single randomly pick example. Since SGD brings fast convergence by randomly choosing one representative example from the data set, it also brings noise between the actual example and its expectation. This noise will make SGD stuck at local optimal point. Then, a Mini-batch gradient descent is proposed as a compromising solution ([Ruder \(2016\)](#)). It chooses a batch size number of examples for each iteration, when the $batch_size = 1$, it turns into an SGD algorithm, when

2.4 NN Application in traffic simulation problem

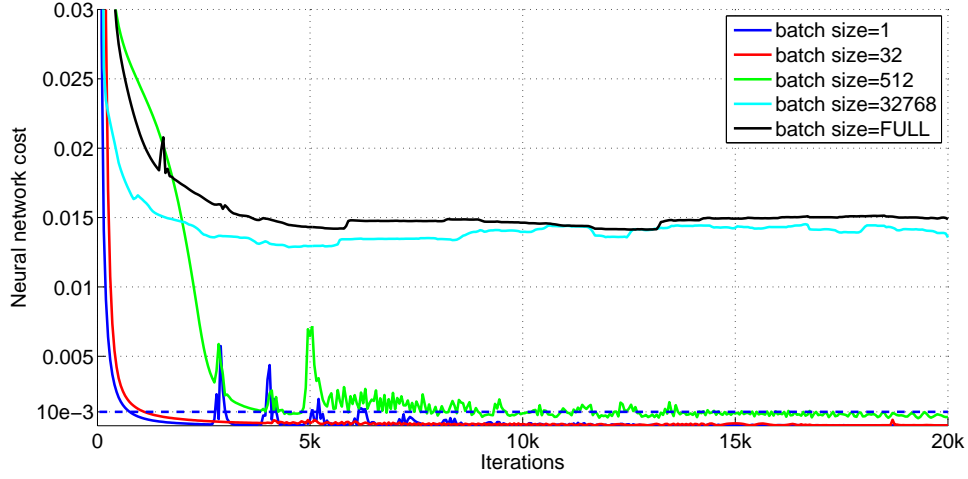


Figure 2.7: Comparison on different Batch Size

the *batch_size* equals to the amount of all examples, it turns into a traditional gradient descent.

Algorithm 3 Procedure of Mini-batch gradient descent learning

- 1: Divide the database by *batch_size*;
- 2: Repeat until convergence: {
- 3: randomly choose a batch *b*:
- 4: for each *i, j* and *l*:

$$w_{ij}^{(l),'} = w_{ij}^{(l)} + \eta \sum_{k=1}^{batch_size_b} \frac{\partial E_k}{\partial w_{ij}^{(l)}} \quad (2.14)$$

- 5: }
-

While training a neural network, the normalization is necessary in the input layer to accelerate the convergence since it unifies the scale of different inputs. Considering training a neural network with deep layers, the gradient could explode or vanish because of the multiple layers. A batch normalization method (Ioffe & Szegedy (2015)) is proposed to accelerate the convergence of a network with such multiple layers. Its idea is to perform a normalization between each hidden layer and the next layer. The variables γ and β are intro-

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

duced for each neuron to represent its mean and standard deviation. so that there will be

$$y^{(k)} = \gamma^{(k)}x^{(k)} + \beta^{(k)} \quad (2.15)$$

γ, β will also be updated respectively for each iteration, so that the feature learned in this layer will not shift.

These tricks on deep learning allow us to train a neural network with more complicated structure.

2.4.3 Learning Process

A basic training procedure of a neural network can be done as in Chapter 2.3, and it is called a Feed-Forward Neural Network (FFNN). While training an actual neural network, in order to avoid overfitting problem, all demonstrations should be separated into two example groups, the training examples, and the testing examples.

As we are about to learn the mobility model from the traffic data, and individual behaviors of a single vehicle are concerned in our simulation, the neural network will not learn by those flow attributes in macroscopic level, such as the traffic flow quantity, traffic density or traffic flow average velocity. Instead, said the attributes extracted from the original database describe the local behavior would be chosen. Either way, the neural network would be built with multiple input and output.

Normally when training an FFNN, the demonstration of examples from both training and testing sets are shuffled out of order, which means that there is not temporal property preserved in the learned model. However, the behavior of vehicles depends on its former behavior in some perspectives. The traffic data captured is sequential data in time. Moreover, for example, when executing an overtaking behavior, the vehicle would change its lane to the left, speed up and then return to its original lane in the right. The whole procedure is a continuous process that usually has been finished together. In order to keep this temporal feature in the mobility model learned by neural network, the recurrent neural network (RNN) should be introduced.

Comparing with FFNN, RNN has a different structure that cycles exist within the network. The cycle setups the association between the output of

2.4 NN Application in traffic simulation problem

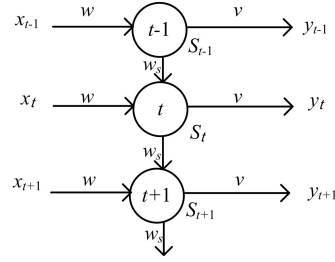


Figure 2.8: Structure of a recurrent neural network

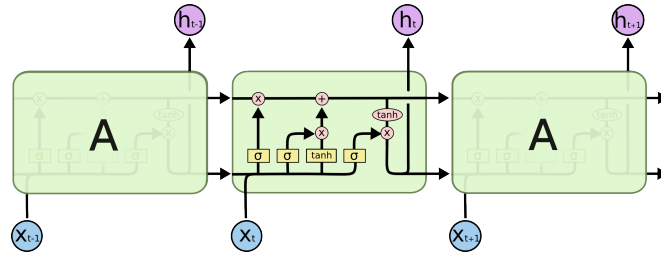


Figure 2.9: Structure of a LSTM network

the current time stamp and the output from the previous time stamp. Thus, the memory property has been carried out and the temporal attributes of the original data have been maintained. RNN has been widely used in taking care of sequential problems, especially for the audio or video sequences ([Graves et al. \(2013\)](#); [Williams & Zipser \(1989\)](#)).

As shown in Figure 2.8, the output at time stamp t is associated with the input at time stamp t , and the state memory of time stamp $t - 1$

$$\mathbf{y}_t = \mathbf{v}f(\mathbf{w}\mathbf{x}_t + \mathbf{w}_s\mathbf{S}_{t-1} + \mathbf{b}_s) + \mathbf{b}_y \quad (2.16)$$

where \mathbf{v} , \mathbf{w} , \mathbf{w}_s are the corresponding weights, \mathbf{b}_s and \mathbf{b}_y are the bias, f is the selected activation function. It can be very difficult to train a standard RNN, because the gradient of error function decays exponentially with time (across the time layers). So that the long short-term memory (LSTM) structure is proposed ([Hochreiter & Schmidhuber \(1997\)](#)). The LSTM is a special kind of RNN that uses special LSTM units to connect input through different time instead of creating the loop directly. Its structure can be found as in Figure 2.9

In detail, the composition of a LSTM unit can be found as in Figure 2.10

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

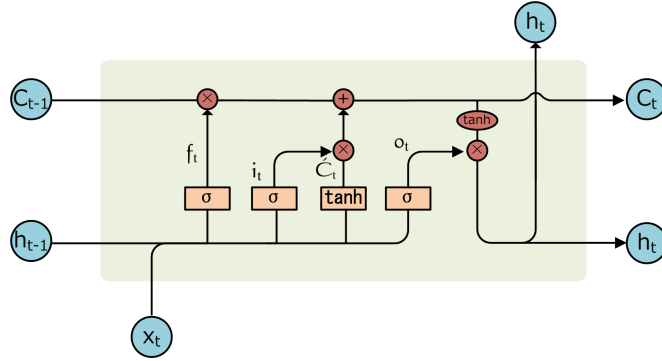


Figure 2.10: Composition of a LSTM unit and its denotations

It seems to be sophisticated since there are three additional gates comparing with the traditional feed forward structure. Suppose that we have input x for single layer in d dimensions and output of single layer h in h dimensions. Then we have:

- input vector: $x \in \mathbb{R}^d$
- output vector: $h \in \mathbb{R}^h$
- bias: $b \in \mathbb{R}^h$
- weights for input x : $W \in \mathbb{R}^{d \times h}$
- weights for output h : $U \in \mathbb{R}^{h \times h}$

The forget gate then as in Equation (2.17) represents whether to keep the value from previous layer or not.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.17)$$

The input gate, which decides what new information would be stored in cell, is as

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.18)$$

Its output combining the new candidate value \tilde{C}_t will go through a tanh layer and be updated to the cell then.

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C) \quad (2.19)$$

2.4 NN Application in traffic simulation problem

The actual updating process can be explained by

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \quad (2.20)$$

where \circ is the hadamard product. A hadamard product of two matrices $\mathbf{A}_{i,j}$ and $\mathbf{B}_{i,j}$ would be $\mathbf{C} = (\mathbf{A} \circ \mathbf{B})_{i,j}$ with each element equals the product of corresponding elements $c_{ij} = a_{ij}b_{ij}$.

There is still an output gate to decide what is about to output.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.21)$$

The finally output will be based on the cell C_t

$$h_t = o_t \circ \tanh(C_t) \quad (2.22)$$

As a conclusion, an LSTM adds three gates to the memory cell of RNN. These three gates will decide the update condition of the cell and as can be seen from Figure 2.9, not only the output h , but also cell memory C passes through layer.

Another thing to be considered, \tanh function is used within an LSTM unit, which is also a commonly used activation function of neural network. Its derivative would be

$$f'(x) = 1 - f(x)^2 \quad (2.23)$$

2.4.4 Implement simulator for virtual traffic flow

As shown in Figure 2.4, it describes the procedure from the virtual vehicle's entry of the simulated scenario to its final animation presentation. However, the flow chart is for a single vehicle, which means there shall exist multiple event loops as described being executed at the same time. Then how to arrange the timing and operation of each vehicle would be a problem to solve at this section.

Firstly, in order to demonstrate our design of the traffic simulation system in a general structural way, a UML-based design is presented. The Unified Modeling Language (UML) is a general-purpose modeling language, which

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

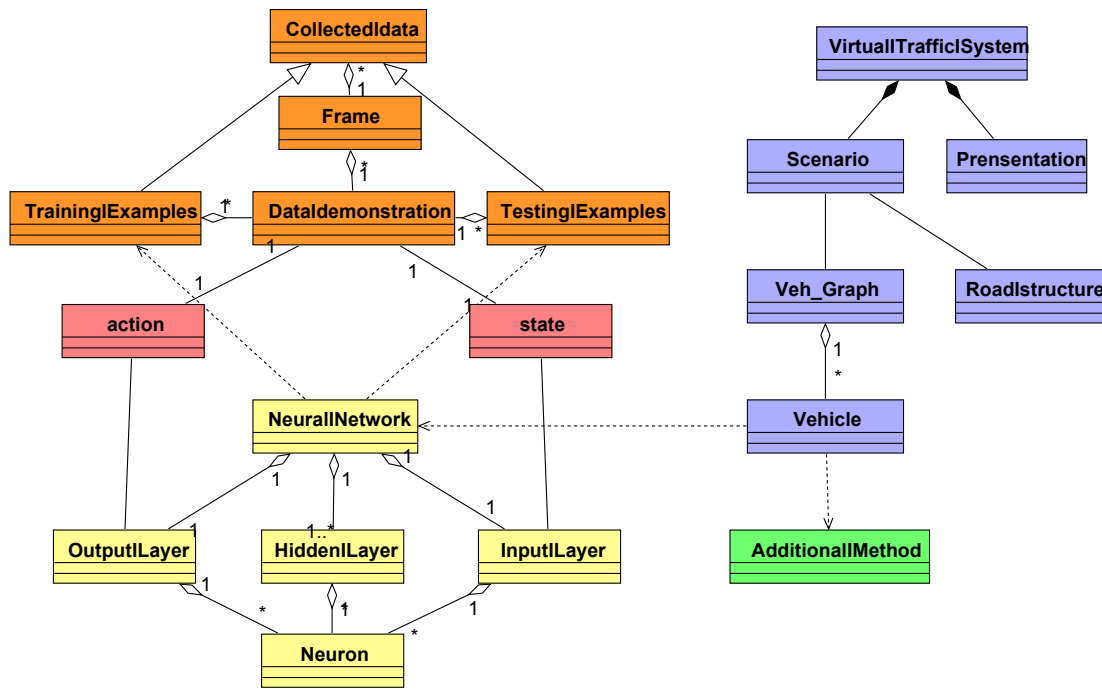


Figure 2.11: Detail System Design in UML Class Diagram

intends to provide a standard way to visualize the design of a system. It contains a large amount of contents to make the design be represented in an intuitive way. In Figure 2.11, the class diagrams are presented to demonstrate the static (structural) view of the design.

Figure 2.11, demonstrates the basic architecture of this design, which describes the multiple kinds of possible relationships between the major parts of the system: original real-world traffic data, neural networks, virtual vehicles, and virtual traffic system. Different parts of the system could be distinguished by colors.

As shown in Figure 2.11, a veh-graph has been established to manage the vehicles. The proposed structure veh-graph includes an index list and a graph $G\langle V, E \rangle$. The time complexity of searching a certain vehicle from the whole vehicle network decreases significantly since we combine the vehicle graph with an index list. The graph is specified as a six-direction graph. The graph contains six directions for each node, which denotes the corresponding relevant location information. Thus, the searching and the calculating cost has been heavily reduced. As the information should be frequently used when we call

2.4 NN Application in traffic simulation problem

the mobility model and form its input, the method with veh-graph would be more efficient.

The establishment and the modification of the graph could use the insert action described in Algorithm 4, and the delete action described in Algorithm 5. Attention should be paid that the established graph is a directed graph. As for a certain node, its precedent of its successor could possibly be another node. So when updating the graph, the operations should be done especially in good order.

As for the presentation of virtual vehicles, which is also known as the animation of traffic flows, there are several existing either commercial or open-source traffic simulation software and platform. For example, the PTV Studio (Mahmud & Town (2016)) is a comprehensive traffic simulation software including sub-software PTV Vissim for microscopic traffic simulation and PTV Visum for macroscopic traffic simulation, etc. In this software group, the user can construct its own virtual environment or load existing virtual event and put their focus on either the change of certain parameters or the global evolution of the whole traffic network. In addition to the commercial software, there also exists the open-source traffic simulation platform, such as SUMO (Krajzewicz *et al.* (2012)). Users of SUMO Simulator can construct a virtual scenario off-line and then perform the generated simulation, while it is also allowed to use a programming language to build and customize a more complicated situation. Novel proposed mobility model, rules, events can then be introduced into the scenario, and by using the interface Traci, the communication between presentation of simulation and its programming logic can be built. Thus, A real-time simulation result can be performed. The interface Traci makes the platform programmable and even more flexible for undefined and uncertain scenarios. It has been setup for the major programming language including C++, Matlab and Python. The animation presentation can be also accomplished by using programming graphics rendering engine, such as the Ogre used in Yu *et al.* (2014b)'s VR-ISSV.

2.4.5 Performances

The data collections used for learning in this thesis is from the database of Next Generation of Simulation (NGSIM), which is collected by the department

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

Algorithm 4 Insert action for a veh-graph

Input:

The roots set $\{r\}$ of current graph $G\langle V, E \rangle$;
The new coming node (vehicle) v_0 ;

Output:

The updated graph $G\langle V, E \rangle$, where $\{V\}$ includes new coming node v_0 and $\{E\}$ includes the edge e_0 ; connected with node v_0 ;

- 1: Find corresponding root r_i according to $v_0.lane$, pointer $p \leftarrow r_i$;
 - 2: **while** $p.suc < v_0$ **do**
 - 3: $p \leftarrow p.suc$;
 - 4: **end while**
 - 5: Setup *precedent* and *successor* of node v_0 :
 $v_0.suc \leftarrow p.suc$; $v_0.pre \leftarrow p$;
 - 6: Setup *precedent* and *successor* of adjacent nodes:
 $p.suc.pre \leftarrow v$; $p.suc \leftarrow v$;
 - 7: **if** Left lane exist **then**
 - 8: Create temporary pointer $temp \leftarrow p.l_pre$;
 - 9: Setup *precedent* and *successor* on left of node v_0 :
 - 10: **while** $temp.pos < v_0.pos$ **do**
 - 11: $v_0.l_pre \leftarrow temp$; $temp \leftarrow temp.suc$;
 - 12: **end while**
 - 13: $v_0.l_suc \leftarrow temp$;
 - 14: Setup *precedent* and *successor* on right of nodes on left:
 - 15: **for** $temp \leftarrow p.l_pre.suc$ to $p.suc.l_pre$ **do**
 - 16: **if** $temp.pos < v_0.pos$ **then**
 - 17: $temp.r_suc \leftarrow v_0$;
 - 18: **else**
 - 19: $temp.r_pre \leftarrow v_0$;
 - 20: **end if**
 - 21: **end for**
 - 22: **end if**
 - 23: Same procedure on the other side.
-

2.4 NN Application in traffic simulation problem

Algorithm 5 Delete action for a veh-graph

Input:

Current graph $G\langle V, E \rangle$, and its corresponding index list **List**;
The under deleting node (vehicle) v_0 ;

Output:

The updated graph $G\langle V, E \rangle$, where $\{V\}$ excludes under deleting node v_0
and $\{E\}$ excludes the edge e_{0j} connected with node v_0 ;

- 1: Locate node v_0 by using the index list **List**;
 - 2: Setup *precedent* and *successor* of adjacent nodes:
 $v_0.pre.suc \leftarrow v_0.suc$; $v_0.suc.pre \leftarrow v_0.pre$;
 - 3: **if** Left lane exist **then**
 - 4: **for** $temp \leftarrow v_0.l_pre$ to $v_0.pre.l_suc$ **do**
 - 5: $temp.r_suc \leftarrow v_0.suc$;
 - 6: **end for**
 - 7: **for** $temp \leftarrow v_0.l_suc$ to $v_0.suc.l_pre$ **do**
 - 8: $temp.r_pre \leftarrow v_0.pre$;
 - 9: **end for**
 - 10: **end if**
 - 11: Same procedure for the other side.
-

of transportation of the United States of America (NGSIM (2017)). Data collections of NGSIM take records on the vehicle's action during a section of highway with seven lanes. The collection contains 15 minutes, 9,809 frames and 1,055,801 queries (each query describes a single vehicle's basic information and its location and velocity in a single frame) of recorded traffic data. Figure 2.12 (top) demonstrates the vehicle distribution throughout time. The quantity of vehicles raises in the first 1000 frames and decreases in the last 1000 frames. Because the observed pool of vehicles has been empty, and as vehicles go into or out of the area, the quantity would reach its peak and keep stable in the middle of the observation.

Our experiments have been setup and run with the processor of 2.50GHz and 8.0GB memory. Matlab has been employed as the experiment environment. The used traffic data from NGSIM has been read into the environment and processed before learning. So the normalized data does not convey any meaning of real-world information directly until it has been scale back to the original scalar. At the first part of our experiments, a basic FFNN is built. As shown in Figure 2.12 (bottom), the curves indicate the accumulation procedure

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

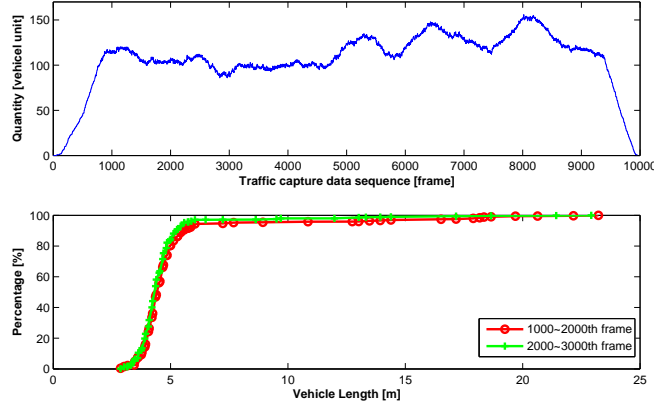


Figure 2.12: Vehicle distribution and vehicle flow composition

of the vehicle quantities by the composition of different type of vehicles, where the red one indicates the situation in the period from frame 1,000 to 2,000, and the green one indicates the frame 2,000 to 3,000. The compositions of vehicle flow are similar however, different in detail. Thus the data from frame 1,000 to 2,000 is chosen as the training example set, while data from frame 2,000 to 3,000 is chosen as the testing example set. According to the selected attributed in our state-action pair, the network has input number at 10 and output number at 3 ($v, a, lane$). The learning rate η is chosen at 0.05 at start. The learning process stopped at the 8000th iteration. The result is roughly measure by MSE, where $E_{training} = 3.917 \times 10^{-4}$ and $E_{testing} = 6.193 \times 10^{-4}$.

Then it comes the second part of our experiments. An RNN is constructed with recovered sequential data from the original database. The same as described above, the basic network of RNN has an input number at 3 and output number at 2 with 10 hidden cells. Initialized learning rate is 0.006, the temporal dependency is set as 5 time steps. Batch size is chosen as 512. The training process can be found as shown in Figure 2.13 after 20,000 times iterations. Cost is measured by entropy.

A perturbation test has also been done in this section to verify the robustness of the proposed method. As shown in Figure 2.14, the result has been achieved by mixing with a Gaussian noise with different input, which is the velocity, the acceleration, and the distances, respectively. The noise is at the amplitude of 5% before the 150th frame, and 10% after the 150th frame. The

2.4 NN Application in traffic simulation problem

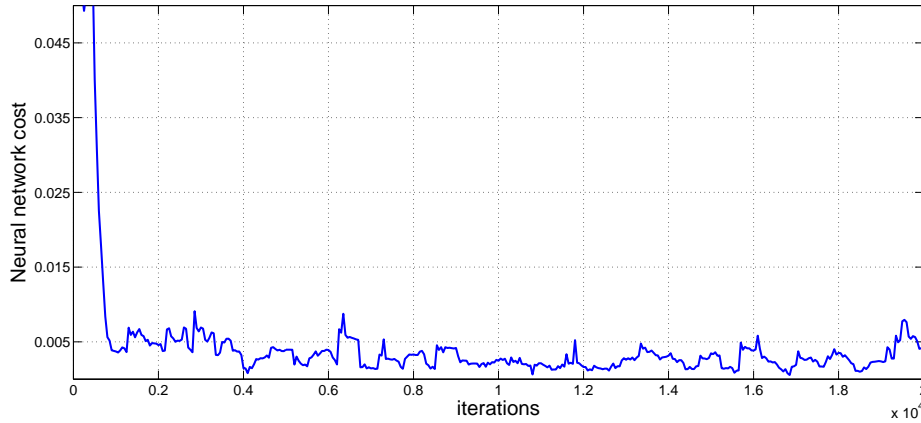


Figure 2.13: Neural network training, result presented by tensor-board, which indicates the changes on cost through iterations

perturbation test is necessary since the real-world drivers could make mistakes in estimating the distance to the preceding vehicle, error and noise exists in communication between vehicles while using V2V or V2X infrastructures. The perturbation test could well imitate those situations and we can see whether the model is robust enough in a real-world application. As shown in Figure 2.14, slightly oscillation happens in the model with a perturbation in acceleration. The performance is averaging at 8×10^{-2} as MSE for perturbation in acceleration. The result is quite as we expected since the acceleration is the higher prime of the velocity and the distance. It would be more sensible on the perturbations on acceleration than the velocity and distance. However, in a real scenario, the drivers would like to tend to estimate only the relative distance to the other vehicles. The local state of the own vehicle (velocity and acceleration) would be achieved by observing the equipped meters, which means the perturbation on distance should be the major concerns. Thus, the learned model is robust guaranteed.

Besides the results of learning the model with neural network from theoretical validation, an actual simulation is also presented in this section, as shown in Figure 2.15. The under simulated scenario is a sudden stop script, which exists in the original data collection. Vehicles (marked in yellow) travels in a period of a 3-lane highway. Since this period of highway is near a toll station, these three lanes are functionally equal. A vehicle (marked in blue) suddenly

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

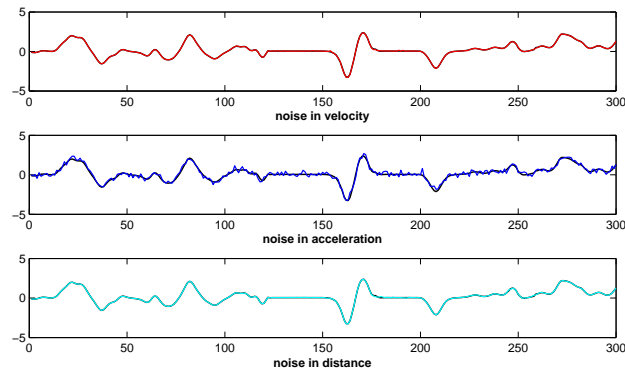


Figure 2.14: Output with Gaussian noise in velocity, acceleration and distance

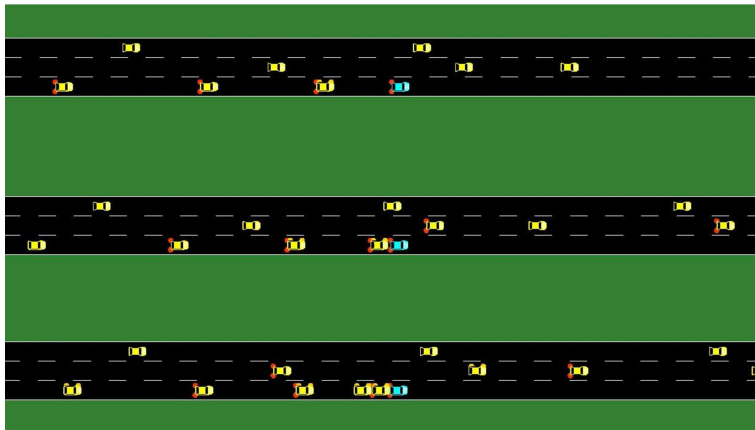


Figure 2.15: A simulation for sudden stop scenario

malfunction in driving and it perform a sudden stop. Then other vehicles in this scenario will interact with this malfunctioned vehicle and behave under control of the mobility model learned by the proposed method.

We can see that the vehicles following the target vehicle decelerate since they have observed that the preceding vehicle has gone wrong. They would perform a lane-change action if the space in the left lane permit to do so. As we can see, the whole process seems natural and realistic, which indicates the effectiveness of our method.

2.5 Discussion and Conclusion

2.5.1 Discussion

Traffic happens mainly in two kinds of scenarios, the highway traffic and the urban traffic. Different properties are shown in these two kinds of scenarios. Highway traffic contains more frequently lane changing behavior, while the speed changes more in the urban traffic because of the existence of the intersections and speed limits.

The example-based method depends on the data collection, which makes the selection of data become more important. Most of the existing data collection of urban traffic that could be checked from the traffic bureau locates in macro-level, such as the count of traffic volumes. Consequently, previous works focused on the global speed prediction. Handling micro-level traffic data would be a complex and messy work. The micro-level data could be found by the image data produced by surveillance cameras, which are mostly set near intersections. It makes behaviors between the two intersections hard to be recalled. Thus, we tend to use the traffic data from highway in this chapter. However, no matter which kind of section the data describes, highway traffic or urban traffic, the learned model would express its characteristic.

In addition, there are also abnormal data in the selected collection. Since relative attributes are considered in learning the model, queries from abnormal vehicles cannot be simply erased, otherwise the vehicles related would be linked to a not-existing vehicle. In the other hand, the preserved queries bring error for the overall learning procedure. Thus, a well-produced data collection would increase the learning result one step further. That would make the curves in [Figure.2.1](#) even more closer. But, at the same time, over-fitting is always a problem we shall face while using a neural network. The used data should be carefully selected, and not redundant.

It should also be noted that this chapter describes the method to implement a virtual traffic simulation. The further applications could be built upon the proposed method. The simulation is of great importance in discovering the latent rule that hidden from the presence.

2. NEURAL NETWORK BASED DATA-DRIVEN MOBILITY MODEL

2.5.2 Conclusion

In this chapter, we briefly introduce a method to produce reliable virtual traffic flow with the proposed novel mobility model. The mobility model comes from the observed real-world traffic data, and is learned by a neural network. Differs from prediction in macro-level, detailed behavior of vehicles is considered.

Comparisons have been constructed between the rule-based method and the example-based method to demonstrate the advantage of using an example-based method to achieve the motion pattern of a vehicle. Since the traffic flow is complicated and sensible facing some unknown traffic situations, the mobility model is more reliable and produces more realistic traffic data than using a conventional car-following mode.

A graph structure is also presented to manage and organize the vehicles that make the proposed model be executed efficiently. Algorithms on updating this graph is given to ensure there is no missing or redundant counting vehicles during simulation.

Finally, the presentation is implemented by employing the SUMO simulator. This technique could further used to help to design and test global traffic control strategy. Quick response of local behaviors could be achieved after some global strategies are made and executed.

Limited by the employed data collection NGSIM as discussed, the learned mobility model can only represent the vehicles in highway with few lane changing actions, which happens to match the demands of a tunnel scenario also. The learned flow could be suitable for urban intersection traffic if corresponding data is used for learning. However, those kinds of data are not easy to get, which could be one major focus in the future.

While running a simulation, the error accumulates over time and it need to be removed by every single interval of time. In this chapter, RNN is used to recall the temporal dependence of the original data. The spatial dependence is expressed by distances to nearby vehicles which could be ameliorated by using a more proper model, which we will discuss in the following chapters.

Chapter 3

Enhanced Mobility Model with HMM

Contents

3.1	Introduction	54
3.2	Traditional Mobility Models and their adaptation	54
3.2.1	Mobility Models	55
3.2.2	Basic adaptation in 1-dimension	62
3.2.3	Adaptation of Car-following model in 2-dimension	69
3.2.4	Summary on Adaptation method of Mobility Models	76
3.3	HMM for Mobility Model improvements	77
3.3.1	HMM in intelligent traffic	80
3.3.2	HMM: Basic knowledge and its denotation	80
3.3.3	HMM combined Mobility Model	87
3.3.4	Performance and result	95
3.4	Discussion and Conclusion	99
3.4.1	Discussion	99
3.4.2	Conclusion	99

3.1 Introduction

In this chapter, enhancement with Hidden Markov Model (HMM) on mobility model proposed in previous Chapter 2 will be discussed. But in order to introduce HMM into our model, adaptations of traditional mobility models are necessary. So an adaptation algorithm is proposed in the first part of this chapter (Chapter 3.2), and it is applied with HMM to enhance our model in the second part (Chapter 3.3).

3.2 Traditional Mobility Models and their adaptation

Generating realistic convincing vehicle trajectories for traffic simulation is a long-time procedure. In the previous chapter, we investigated that a learning-based method using a neural network can impact this domain. However, there is still a lot of work to do to obtain a realistic enough model. In fact, in order to ameliorate the result one step further, it is necessary to know the related works on the domain of conditional mobility models in detail, which will bring us a reasonable fashion to summarize. Thus, the historical vehicle model, or to be more precisely, the vehicular mobility model will be located as the major concern of this chapter.

Vehicular Mobility Model (VMM) comes from the concept of Mobility Model (MM) in the Mobile Ad hoc Network (MANET). The MANET is a collection of wireless nodes communicating with each other. Using the principles of MANET and adding a little more specifications, the concept of Vehicular Ad hoc Network (VANET) comes into being, which is specified in studying the movement of vehicles in intelligent transportation. The VMM is designed to describe the movement pattern of vehicles, and the fact that how their local properties, which normally indicate their location, velocity, and acceleration, change over time. A good VMM should attempt to mimic the behaviors of real vehicles.

Usually, the various VMMs (or MMs) can be of huge differences with each other, since they may focus on reflecting different real-world properties. So that, in this chapter, the basic categories of MM will be presented first. Despite

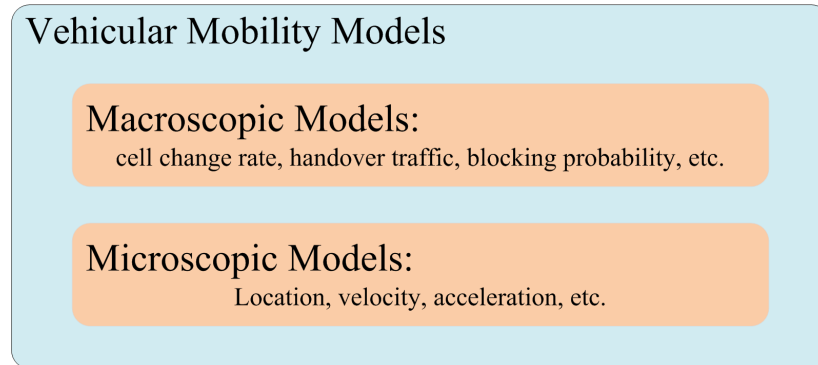


Figure 3.1: Categorization on Mobility Models: scale

the diversity of real-world properties, the temporal-spatial relationships hold an important weight. We will introduce the traditional expression of temporal dependency and spatial dependency in VMMs. Besides, adaptation on VMM will be necessary, so that an adaptation method will be given in detail in both 1 - dimension and 2 - dimension.

3.2.1 Mobility Models

3.2.1.1 Basic categories on Mobility Models

Similarly to the categories on researches on transportation, the VMMs have been built both in macroscopic level, in which the researchers focus on the macroscopic features, such as the movement of relative to a particular area including the cell change rate, handover traffic and blocking probability, etc., and in microscopic level that the models are mainly formulated by vehicles' location, velocity and acceleration, as shown in Figure 3.1.

[Harri et al. \(2009\)](#) has investigated various of mobility models and made comparisons between them according to the structure employed and features concerned in macroscopic level and the represented kinds of behaviors in microscopic level. There exist multiple other categorizations, such as the categorization as in ([Bai & Helmy \(2004\)](#)), it could be categorized into the trace-driven mobility models and synthetic mobility models. A categorization based on the specific restrictions on movement could also be given, thus, there would be temporal mobility models, in which the movement is affected by its historical movement, spatial mobility models, in which the correlated manners are

3. ENHANCED MOBILITY MODEL WITH HMM

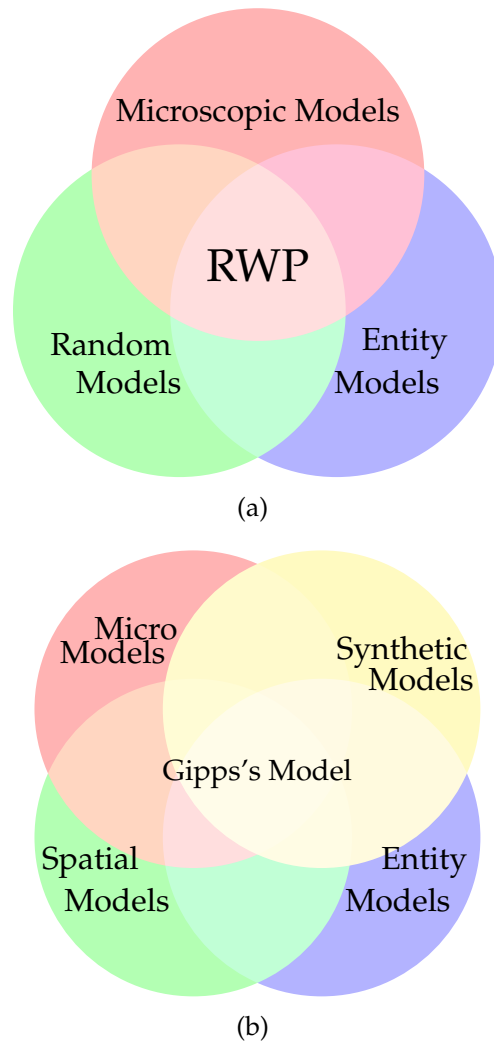


Figure 3.2: Category Overlapping on Mobility Models

considered, and random mobility models, which brings randomness into the models. While according to (Camp *et al.* (2002)), there could be entity mobility models and group mobility models that depend on whether the actions are completely independent of each other. These categorizations is based on different point of view, so that there may be overlapping between different categorizations. For example, the most simple and frequently used random waypoint model could be a microscopic model, meanwhile a random mobility model, yet an entity mobility model, as shown in Figure 3.2.

As can be drawn from Figure 3.2(a), the traditional RWP model focuses on the detailed movement of single individual entities, so that it is a micro-

3.2 Traditional Mobility Models and their adaptation

scopic model. Meanwhile, there are normally not any group properties or behaviors that are expressed by RWP, so that it is also an entity model. At last, it is a random model since the future plan is decided quite arbitrary. A Gipps's car-following model (Gipps (1981)) as shown in Figure 3.2(b), which will be presented in detail later in the following section, concerns the acceleration controller, the deviations on distance and velocity between the car and its preceding one. Since all these factors are in microscopic level, the model itself is a microscopic model. It is also a VMM with spatial dependency restrictions since the relative distance and velocity deviations are considered. Although there are properties from multiple vehicles in this model, there is no correlated behavior. The relative things between these vehicles are only helped to decide the future acceleration of the following vehicle, and it does not affect the preceding vehicle back. Synthetic model means Gipps's car-following does not come from real world tracing, on the contrary, there is a synthetic expression of this model.

3.2.1.2 Movement restrictions in Mobility Models

Despite the diversity on categorizations, the movement of the vehicle is the core part of the VMMs. Different models implicit this character differently. For example, the easiest and most frequently used "Random WayPoint model" (RWP) Broch *et al.* (1998), along with the random walk model Davies *et al.* (2000); Zonoozi & Dassanayake (1997) and random direction model Royer *et al.* (2001) determine the future velocity randomly, and the future direction turning randomly, which will locate the future movement in a random fashion.

Restrictions are then added to precise the movement and these different restrictions are the basic component that distinguishes different models. In the basic RWP, extreme behaviors including a sudden stop or sharp turn happen frequently due to the lack of corresponding restrictions. Some adjustments should be made since the differences between MANET models and VANET models that vehicles move in a lower degree of freedom than the mobile nodes in MANET, such as pedestrians. So that the physical restrictions of vehicles should be considered, that vehicle cannot move exceed its physical capability: it cannot accelerate or decelerate over its maximal acceleration or deceleration;

3. ENHANCED MOBILITY MODEL WITH HMM

it cannot maintain a stable run if it has already exceeded the maximal safety velocity.

In addition to the physical capability restrictions, there are others such as the artificial restrictions, which is used to characterize some behaviors in detail manually, and it is usually known as the temporal dependency restrictions and the spatial dependency restrictions.

As for the Gauss-Markov Mobility Model introduced by [Liang & Haas \(1999\)](#), it could be presented in the following form:

$$\begin{aligned} v_t^x &= \alpha v_{t-1}^x + (1 - \alpha)v_d^x + \sigma\sqrt{1 - \alpha^2}w_{t-1}^x \\ v_t^y &= \alpha v_{t-1}^y + (1 - \alpha)v_d^y + \sigma\sqrt{1 - \alpha^2}w_{t-1}^y \end{aligned} \quad (3.1)$$

where $v = \{v_d^x, v_d^y\}$ is the drift velocity preset, α indicates the memory level of this model and $w = \{w^x, w^y\}$ is the Gaussian random variable. It can be noticed that the velocity $v_t = \{v_t^x, v_t^y\}$ at time t is dependent on the velocity $v_{t-1} = \{v_{t-1}^x, v_{t-1}^y\}$ at time $t - 1$. The model has the memory on velocity at the previous time slot and this previous velocity will help to decide the future velocity. Therefore the Gauss-Markov model is a temporally dependent mobility model. Meanwhile, if the memory parameter $\alpha = 0$, the model turns to be memoryless, and then Equation (3.1) is

$$\begin{aligned} v_t^x &= v_d^x + \sigma w_{t-1}^x \\ v_t^y &= v_d^y + \sigma w_{t-1}^y \end{aligned} \quad (3.2)$$

which is a Random Walk Model. If $\alpha = 1$, then we have

$$\begin{aligned} v_t^x &= v_{t-1}^x \\ v_t^y &= v_{t-1}^y \end{aligned} \quad (3.3)$$

which is called a fluid flow model in traffic theory.

A mobility model with temporal dependency restrictions recalls its previous value (velocity, as in the example of Gauss-Markov model, yet other attributes could be used in a different model) to decide its future value. The memory level is usually decided by a parameter α to adjust the degrees of memory. The temporal dependency restrictions can be used to fulfill some memorial features and to smooth the vehicle's behaviors.

3.2 Traditional Mobility Models and their adaptation

While in order to fulfill other correlated behaviors, it is needed to add the spatial dependency restrictions. For example, the Reference Point Group Mobility Model (RPGM) [Hong *et al.* \(1999\)](#) uses reference points to represent the relative group movement to the individual movement and a logical center for the group is used to calculate the group movement. Individual mobile entities move about their own reference points and their locations are updated according to the group's logical center. The reference points and their relative location to the logical center is fixed, once the logical center's position is updated, the new reference points locations are then drawn, and individual random movements will be pulled toward their new reference points locations. The RPGM depicts the group behavior among multiple mobile entities, and there are also other spatial dependency restrictions to achieve the multi-functional roles among one group, such as the Leader-Follower relationship among pedestrian groups. The leader leads the group and will have attractions to the group members, while the followers follow the lead. Members from different groups may have a repulsive effect to each other if their targets are in opposite directions. As for vehicles, the car-following among vehicle platoon driving also benefits from the spatial dependency restriction.

3.2.1.3 Adaptation to Probabilistic Models

So far, a brief introduction has been given to the traditional mobility models, including the categorizations and their corresponding inner logic, the major representatives of temporal dependency restrictions and spatial dependency restrictions. To be noticed that the VMMs are defined to describe how the vehicular properties change over time. Whether or not the model is temporal dependent, the model is usually presented by time related function. Yet it could be expressed in another way, we call it a probabilistic represented mobility model, which using the probabilities distribution of different possible performance to characterize the model.

To distinguish the mobility model based on a probability distribution and the probabilistic represented mobility model we named it here, one thing should be clarified. The former is an MM which uses a probability distribution to help to decide the future velocity (or other attributes). There is an uncertain component within the model, but the model itself is still represented by variables,

3. ENHANCED MOBILITY MODEL WITH HMM

factors or attributes that make physical senses. For example, if we choose a Gaussian distribution to determine the relative changes of velocity in an RWP, then the RWP is a mobility model based on a Gaussian distribution. It is still in shape of a normal VMM, which we can find out physical understanding of its representation.

[Kendziorra *et al.* \(2016\)](#) gives another example on the mobility model that based on a probability distribution. A data-driven stochastic car-following model is presented. The target acceleration a of the following vehicle is modeled as drawn from a probability distribution that is sampled directly from the data. The Laplace distribution is claimed to be the best fit according to their modeling, as

$$L_{\mu,\sigma}(a) = \frac{1}{2\sigma} e^{-|a-\mu|/\sigma} \quad (3.4)$$

where μ and σ are the parameters of distribution. Data set of the naturalistic driving study ([Fancher \(1998\)](#)) is used to determine these distribution parameters. It is obvious that the distribution parameter in ([Kendziorra *et al.* \(2016\)](#)) is in need of calibration by the real-world observed data. In fact, there is a whole set of procedure in data processing in their work.

While the latter is a VMM in the formation of probability representation. To be more generally speaking, considering about the format of output of a model, it could be either discrete or continuous, then there would be a different way to characterize the corresponding distribution: the probability mass function (pmf) is used to the discrete situation and usually the probability density function (pdf) is employed to the continuous situation. The pmf or pdf describes the probability of certain output will be observed. A traditional mobility model is possible to be adapted into this representation. The flow chart of the adaptation algorithm can be found as in [Figure 3.3](#).

Differences between these two concepts have been made clear, that the emphasis has been plated on the representation form of a VMM. The core concept to distinguish can be seen as in [Figure 3.4](#), in which v stands for the velocity and S stands for the spacing. subscript v_f and S_f indicate the property of the following vehicle and subscript v_p and S_p indicate the property of preceding vehicle. v' and S' stands for the future out of the model after a time τ . A model in its dynamics representation can be expressed as finding the corresponding output according to the various input; on the contrary, a model

3.2 Traditional Mobility Models and their adaptation

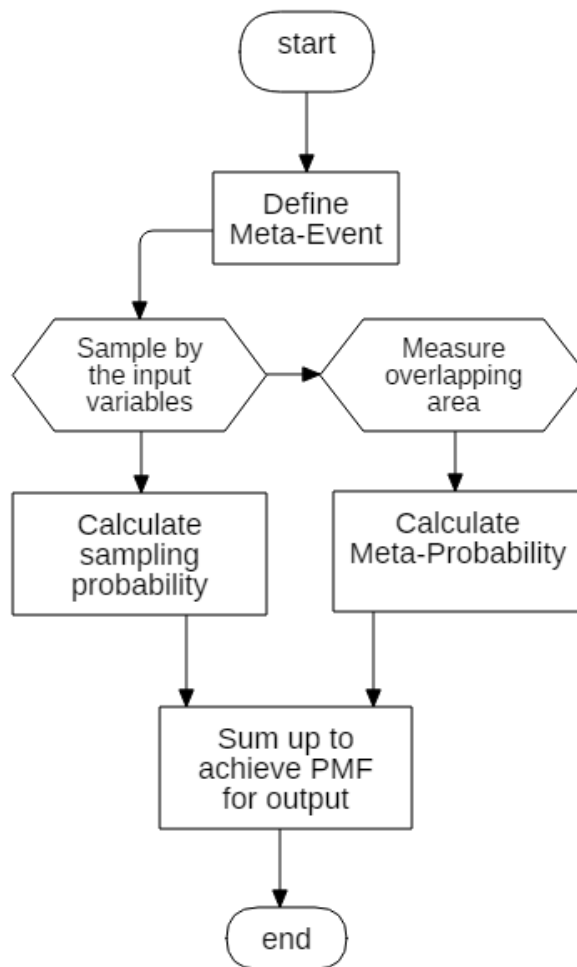


Figure 3.3: Flow chart of the adaptation algorithm

3. ENHANCED MOBILITY MODEL WITH HMM

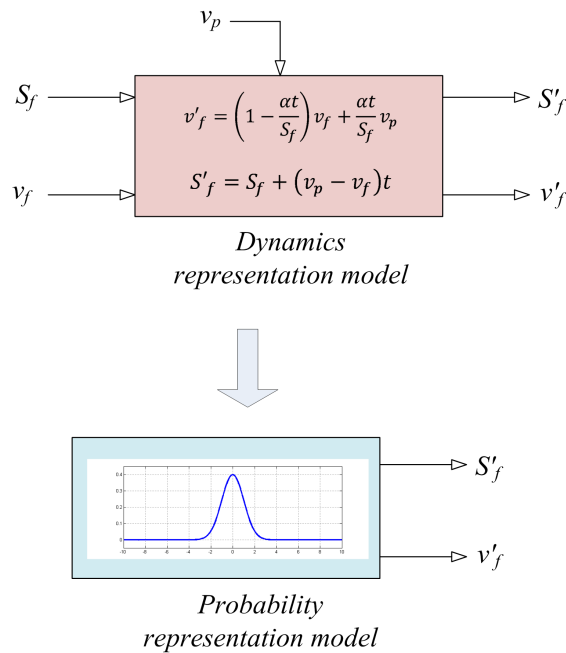


Figure 3.4: Adaptation of representation of a model

in its probability representation can be expressed as the possible probability that we can observe a certain output of this model. Attention should be paid that it is possible that one single VMM can have both forms of representations: its normal representation, for example by the kinetic equations, and a corresponding probabilistic representation. If an adaptation method between these different representations for one VMM is available (as in Figure 3.4), then we could achieve almost double time model than before. By all means, it would then broaden the scope of the application on such a VMM.

To be specified in the following chapter, a probabilistic representation fits more with the data-driven machine learning based methods. Thus, it will lead to another combination between different research areas.

3.2.2 Basic adaptation in 1-dimension

The adaptation method of a VMM to in 1-dimension is given in this section. We can get a basic knowledge of the general idea behind this adaptation method by studying the 1-dimension situation. Moreover, a more practical adaptation example in 2-dimension situation, which is more commonly used in practice

3.2 Traditional Mobility Models and their adaptation

will be given in the following section.

The basic demand of this adaptation is to convert the VMM into a more proper and more compatible form with a machine learning probabilistic model, such as a Hidden Markov Model being used in the following chapter. A traditional VMM with its dynamics form cannot be directly used in an HMM since the model in HMM is described by an emission matrix of its observation which is a probabilistic model.

To start with, a VMM in its dynamics form will be regarded as the input of the adaptation. A car-following model would be good, which is the most commonly used, simple, meanwhile representative as to all the VMMs. According to the history of the development of car following models, we would like to use the one as described in Equation (3.5), as

$$a_n(t + \tau) = \alpha[v_{n-1}(t) - v_n(t)] \quad (3.5)$$

where $a_n(t)$ and $v_n(t)$ stand for the acceleration and velocity of the vehicle No. n at time t . τ is the minimal reaction time, which can also be used to measure the step length. α is the parameter to control the magnitude of the model.

According to this car-following model (Gipps (1981)) in Equation (3.5), the acceleration of the following vehicle is only related to the deviation of its velocity to ones of preceding vehicle. By the typical kinetic equations, the whole representation of the following vehicle including velocity and location can be calculated.

$$\begin{aligned} a_n(t_0 + \Delta t) &= \alpha[v_{n-1}(t_0) - v_n(t_0)] \\ v_n(t_0 + \Delta t) &= v_n(t_0) + \alpha[v_{n-1}(t_0) - v_n(t_0)]\Delta t \\ p_n(t_0 + \Delta t) &= p_n(t_0) + \int_{t_0}^{t_0+\Delta t} v_n(t) \mathbf{d}t \end{aligned} \quad (3.6)$$

To be noticed that as the τ in Equation (3.5), the Δt stands for the interval of updating the basic representation, so that it is assumed that the movement within one interval follows the linear motion laws. Therefore, it could be calculated as in Equation (3.6), which means all the other attributes besides velocity can be represented by the velocities, such as $a = f(v)$ and $p = g(v)$.

Then, we can focus on the iteration of velocity through time, because it can create a close loop for its update. Emission of an HMM can be both con-

3. ENHANCED MOBILITY MODEL WITH HMM

tinuous and discrete. Since discrete emission will be used in the following chapter, a discrete model is employed here for demonstration. So that as for a velocity v , its evaluation has been discretized into M intervals by a length d : V_1, V_2, \dots, V_M , where we have interval $V_i : [v_i, v_i + d]$

If the following velocity of vehicle at current time is denoted as v_f , its future velocity after a time interval t is denoted as v' , the velocity of its preceding vehicle is denoted as v_p . we have

$$\begin{aligned} v' &= v_f + at \\ &= v_f + \alpha(v_p - v_f)t \\ &= (1 - \alpha t)v_f + \alpha t \cdot v_p \end{aligned} \quad (3.7)$$

Our target is to achieve a probability distribution of all M possible discrete intervals.

$$P_m = P(v' | v' \in V_m) \quad , m \in \mathbb{N}^+, |m| \leq M \quad (3.8)$$

According to the law of total probability, if we sample by the velocity of previous time slot,

$$P_m = \sum_{i=1}^M P(v' \in V_m | v_f \in V_i) P(v_f \in V_i) \quad (3.9)$$

Equation (3.9) means that the probability of future velocity v' locating on the m th interval is the sum of all conditional probabilities when given the velocity of following vehicle in the previous time slot. One step further, if we sample by the velocity of preceding vehicle,

$$\begin{aligned} P_m &= \sum_{i=1}^M P(v' \in V_m | v_f \in V_i) P(v_f \in V_i) \\ &= \sum_{i=1}^M \sum_{j=1}^M P(v' \in V_m | v_f \in V_i, v_p \in V_j) P(v_p \in V_j) P(v_f \in V_i) \end{aligned} \quad (3.10)$$

Considering about the monotonic property of Equation (3.7), there are two different situations with either $\alpha t > 1$ or $\alpha t \leq 1$.

Situation 1 $\alpha t > 1$

then $1 - \alpha t < 0$, according to its monotonic property, the evaluation interval of

3.2 Traditional Mobility Models and their adaptation

v' would be:

$$\begin{aligned} v' &\leq v'_{max} = (1 - \alpha t)v_{f_{min}} + \alpha t \cdot v_{p_{max}} \\ v' &\geq v'_{min} = (1 - \alpha t)v_{f_{max}} + \alpha t \cdot v_{p_{min}} \end{aligned} \quad (3.11)$$

Suppose $v_f \in V_i$ and $v_p \in V_j$, meanwhile $j = i + k, k \in \mathbb{Z}$ and $|k| \leq M$. Equation (3.11) would be

$$\begin{aligned} v' &\leq v'_{max} = (1 - \alpha t)v_i + \alpha t(v_j + d) \\ &= v_i + \alpha t(k + 1)d \\ v' &\geq v'_{min} = (1 - \alpha t)(v_i + d) + \alpha t \cdot v_j \\ &= v_i + \alpha t(k - 1)d + d \end{aligned} \quad (3.12)$$

Length of the possible evaluation interval of v' is

$$l = v'_{max} - v'_{min} = (2\alpha t - 1)d \quad (3.13)$$

Given the evaluation interval V_i , by comparing the value of v'_{min} and v_j , the relative location of interval V_i, V_j and $[v'_{min}, v'_{max}]$ can be achieved:

$$v'_{min} - v_j = (\alpha t - 1)(k - 1)d \quad (3.14)$$

It can be seen as shown in Figure 3.5, as $\alpha t > 1$

- when $k > 1$, we have $v'_{min} > v_j$;
- when $k = 1$, we have $v'_{min} = v_j$;
- when $k < 1$, we have $v'_{min} < v_j$.

Specially, if comparing v'_{max} and v'_j , we have

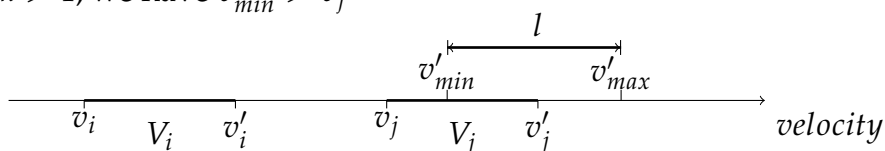
$$v'_{max} - v'_j = v'_{max} - (v_j + d) = (\alpha t - 1)(k + 1)d \quad (3.15)$$

- when $k = -1$, we have $v'_{max} = v'_j$, which is $v'_{max} = v_i$.

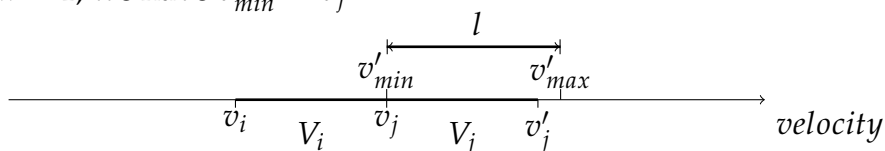
Locational relationship of these intervals can help us calculate the overlapping length of $[v'_{min}, v'_{max}]$ and an certain interval, which indicates the corresponding frequency of transition to this interval. $l_{in}^{(j)}$ is used to represent this overlapping length when v_f is in interval V_i , v_p is in V_j and v' is in V_m , and $l_{max},$

3. ENHANCED MOBILITY MODEL WITH HMM

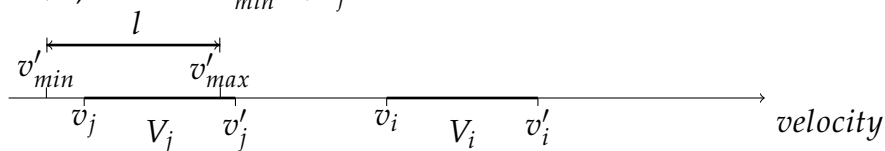
(1) when $k > 1$, we have $v'_{min} > v_j$



(2) when $k = 1$, we have $v'_{min} = v_j$



(3) when $k < 1$, we have $v'_{min} < v_j$



(4) when $k = -1$, we have $v'_{max} = v_j + d$

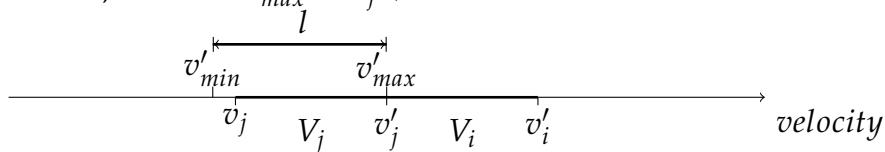


Figure 3.5: Schematic diagram on the location of V_i, V_j , and possible evaluation interval of v'

3.2 Traditional Mobility Models and their adaptation

l_{min} are used to represent the boundary of target interval. Then the possible conditions are listed as below.

- if $v'_{min} > l_{max}$, then $l_{im}^{(j)} = 0$;
- else if $v'_{max} > l_{max}$,
 - when $v'_{min} \geq l_{min}$, then $l_{im}^{(j)} = l_{max} - v'_{min}$;
 - when $v'_{min} < l_{min}$, then $l_{im}^{(j)} = d$;
- else if $v'_{max} < l_{max}$, then $l_{im}^{(j)} = v'_{max} - l_{min}$;
- else if $v'_{max} < l_{min}$, then $l_{im}^{(j)} = 0$.

It can also be seen as in Figure 3.6. We can summarize it into one equation:

$$l_{im}^{(j)} = \max\{\min\{l_{max}, v'_{max}\} - \max\{l_{min}, v'_{min}\}, 0\} \quad (3.16)$$

Situation 2 $\alpha t \leq 1$

then $1 - \alpha t \geq 0$. Attention should be paid that coefficient of v_f and v_p now meet the following conditions:

$$(1 - \alpha t) + \alpha t = 1 \quad (3.17)$$

and

$$\begin{aligned} (1 - \alpha t) &\in [0, 1] \\ \alpha t &\in [0, 1] \end{aligned} \quad (3.18)$$

Which means v' is an interpolation of v_f and v_p . Comparing with the previous situation, v' has different monotonic property, the new evaluation interval of v' would be:

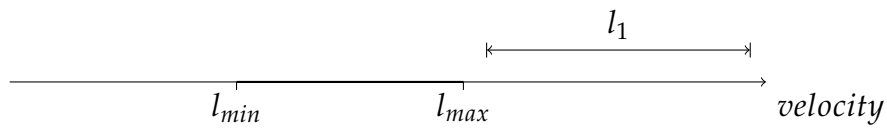
$$\begin{aligned} v' \leq v'_{max} &= (1 - \alpha t)v_{f_{max}} + \alpha t \cdot v_{p_{max}} \\ &= v_i + d + k\alpha t d \\ v' \geq v'_{min} &= (1 - \alpha t)v_{f_{min}} + \alpha t \cdot v_{p_{min}} \\ &= v_i + k\alpha t d \end{aligned} \quad (3.19)$$

Length of the possible evaluation interval of v' changes to

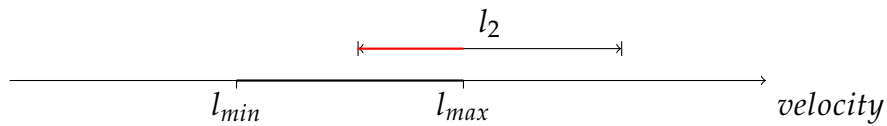
$$l = v'_{max} - v'_{min} = d \quad (3.20)$$

3. ENHANCED MOBILITY MODEL WITH HMM

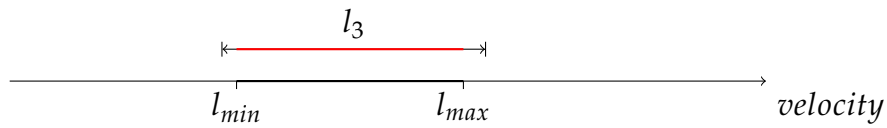
(1) No overlapping area between l_1 and $[l_{min}, l_{max}]$, so that $l_{im}^{(j)} = 0$



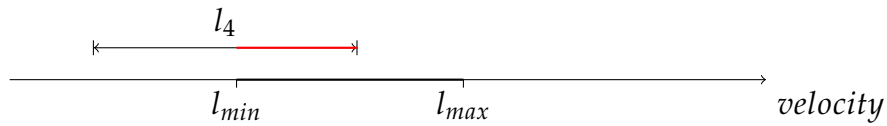
(2) Overlapping area as shown in red, $l_{im}^{(j)} = l_{max} - v'_{min}$



(3) Overlapping area fully covers $[l_{min}, l_{max}]$ as shown in red, $l_{im}^{(j)} = d$



(4) Overlapping area as shown in red, $l_{im}^{(j)} = v'_{max} - l_{min}$



(5) No overlapping area between l_5 and $[l_{min}, l_{max}]$, so that $l_{im}^{(j)} = 0$

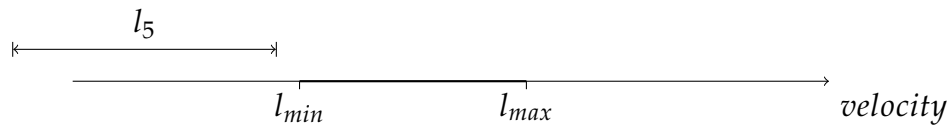


Figure 3.6: Overlapping area demonstration for different $[v'_{min}, v'_{max}]$

3.2 Traditional Mobility Models and their adaptation

Since l has the same length d as each interval V_m , their locational relationship is much simpler than it is in previous situation, the overlapping length $l_{im}^{(j)}$ can be calculated similarly as in Equation (3.16). Practically, t represents the reaction time, and has been evaluated at $t = \frac{2}{3}$ or $t = 2$, while $\alpha = 12$ is the most commonly choice, so normally we have $\alpha t > 1$ as in Situation 1.

The probability $P(v' \in V_m | v_f \in V_i, v_p \in V_j)$ in Equation (3.10) is the core if we want to achieve the distribution. In different situation of αt 's evaluation, l and $l_{im}^{(j)}$ can be always obtained as above. If we denote the probability above as $p_{im}^{(j)}$, according to the physical meaning this probability, then

$$p_{im}^{(j)} = \frac{\text{OverlappingLength}}{\text{IntervalLength}} = \frac{l_{im}^{(j)}}{l} \quad (3.21)$$

where $l_{im}^{(j)}$ and l can be calculated according to Equation (3.13), (3.20) and (3.16). Since the velocity is sampled evenly, there are totally M possible values, so

$$\begin{aligned} P(v_f \in V_i) &= \frac{1}{M} \\ P(v_p \in V_j) &= \frac{1}{M} \end{aligned} \quad (3.22)$$

Thus, Equation (3.10) turns into

$$P_m = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M p_{im}^{(j)} \quad (3.23)$$

Examining P_m 's value for every discrete $m \in [1, M]$, the probability distribution of this car-following is then achieved.

In summary, the adaptation algorithm for 1-dimension can be found as in Algorithm 6.

3.2.3 Adaptation of Car-following model in 2-dimension

When following a preceding vehicle, ignoring the spacing distance is not a proper solution. Differs from the simple 1-dimension car-following model used in previous section, in this example for 2-dimension adaptation, we focus on the most basic meanwhile most commonly used car-following model as in

3. ENHANCED MOBILITY MODEL WITH HMM

Algorithm 6 Adaptation Method in 1-dimension

Input:

Dynamics representation of a car-following model in 1-dimension: $v = f(\mathbf{in})$;

Output:

Probability representation of a car-following model. As for a discrete situation, the probability distribution is described by its probability mass function (pmf): $\mathbf{P}(v = o_i)$.

- 1: Search for a closed iterative loop of v_f and v'_f ;
 - 2: According to the Total Probability Law, Sample by the *input* variables v_f and v_p , until you have the pmf described by the meta probability $\mathbf{P}(v = o_i|v_f, v_p)$;
 - 3: l = full length of evaluation interval;
 - 4: **for** Every possible i , that o_i is in sample space **do**
 - 5: l_i = overlapping length of indicated interval and evaluation interval;
 - 6: Meta probability $\mathbf{P}(o = v_i|v_f, v_p) = \frac{l_i}{l}$;
 - 7: **end for**
 - 8: **for** Each *input* variables in_j : v_f, v_p **do**
 - 9: $\mathbf{P}(in_j)$ can be calculated according to the sampling rule of variable in_j ;
 - 10: **end for**
 - 11: $\mathbf{P}(v = o_i) = \sum_{j_1=1}^{N_1} \sum_{j_n=1}^{N_n} \mathbf{P}(v = o_i|v_f, v_p)\mathbf{P}(v_f = j_1)\mathbf{P}(v_p = j_n)$.
-

3.2 Traditional Mobility Models and their adaptation

Equation (3.24).

$$a_n(t + \tau) = \alpha \frac{[v_{n-1}(t) - v_n(t)]^k}{[s_{n-1}(t) - s_n(t)]^l} v_n^m(t) \quad (3.24)$$

Hence a vehicle under control of this car-following model could be represented by a tuple $\langle v, S \rangle$ in a 2 dimensional space. Combining with typical kinetic equations, we shall have a closed expression of the moving behavior of a vehicle as in Equation (3.25), i.e. the vehicle moves within a $\langle v, S \rangle$ 2-dimension space.

$$\begin{aligned} v_n(t_0 + \Delta t) &= v_n(t_0) + \alpha \frac{v_{n-1}(t_0) - v_n(t_0)}{S_n(t_0)} \Delta t \\ S_n(t_0 + \Delta t) &= S_n(t_0) + [v_{n-1}(t_0) - v_n(t_0)] \Delta t \end{aligned} \quad (3.25)$$

where Equation (3.25) is the simple deduction on typical kinetic model with common evaluation on Equation (3.24): $k = 1, l = 1, m = 0$. As for the vehicle system, only its velocity and spacing are regarded as variables, the velocity of its preceding vehicle is treated as an input that according to a certain preceding velocity, the following iterative relationship can be get:

$$\begin{aligned} v' &= \left(1 - \frac{\alpha t}{S_0}\right) v_f + \frac{\alpha t}{S_0} v_p \\ S' &= S_0 + (v_p - v_f) t \end{aligned} \quad (3.26)$$

where v_f, v_p, v' is as defined before, and S_0 is the spacing to preceding vehicle. So $S_0 > 0$. Normally there is $S_0 > \alpha t$, so that v' could be seen as the interpolation of v and v_0 .

Instead of intervals in one dimension, areas in two dimensions will represent the possible observations, as shown in Figure 3.7.

As for a specified preceding velocity v_p , other inputs of the dynamics model are strict to the area of observation $v_f \in [v_{min}, v_{max}]$ and $S_0 \in [S_{min}, S_{max}]$, then the outputs will locate within an area, boundaries are described by the

3. ENHANCED MOBILITY MODEL WITH HMM

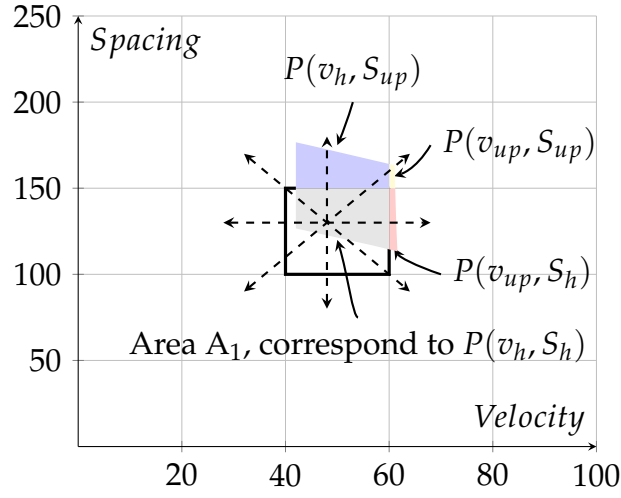


Figure 3.7: "Velocity"- "Spacing" plane in 2-dimension and probability denotations

parametric equations given by Equation (3.27).

$$\begin{aligned}
 \text{upper} : & \begin{cases} v' = (1 - \frac{\alpha t}{S_{max}})v_f + \frac{\alpha t}{S_{max}}v_p \\ S' = S_{max} + (v_p - v_f)t \end{cases}, v_f \in [v_{min}, v_{max}] \\
 \text{lower} : & \begin{cases} v' = (1 - \frac{\alpha t}{S_{min}})v_f + \frac{\alpha t}{S_{min}}v_p \\ S' = S_{min} + (v_p - v_f)t \end{cases}, v_f \in [v_{min}, v_{max}] \\
 \text{left} : & \begin{cases} v' = (1 - \frac{\alpha t}{S_0})v_{min} + \frac{\alpha t}{S_0}v_p \\ S' = S_0 + (v_p - v_{min})t \end{cases}, S_0 \in [S_{min}, S_{max}] \\
 \text{right} : & \begin{cases} v' = (1 - \frac{\alpha t}{S_0})v_{max} + \frac{\alpha t}{S_0}v_p \\ S' = S_0 + (v_p - v_{max})t \end{cases}, S_0 \in [S_{min}, S_{max}]
 \end{aligned} \tag{3.27}$$

The probabilities such as $P(\text{hold})$ in 2 dimensions situation would then be joint event on both v and S . For example,

$$P(\text{hold}) = P(v_{hold}, S_{hold}) = \frac{A_1}{\sum_i A_i} = \frac{\iint_{A_1} dv dS}{\iint_{\sum_i A_i} dv dS} \tag{3.28}$$

where A_1 and A_i stands for the area as shown in Figure 3.7. They can be obtained by calculating the integral on corresponding area. *hold* is the symbol of a transition event that means the corresponding variable does not change its evaluation interval.

Figure 3.8 demonstrates the impact on different speed of preceding vehicle

3.2 Traditional Mobility Models and their adaptation

v_p . Instead of an ideal model, vehicle's mobility may bring limitation on its behavior, which means some of the control results that go beyond vehicle's capability will be corrected with the maximum acceleration a_{max} and maximum deceleration b_{max} . As shown in Figure 3.8, vehicle could not be described only by the certain area marked in dashed orange, which means the vehicle cannot accelerate to such a speed within the reaction time. So the actual parametric equations will be given by:

$$\begin{cases} v' = \max\{\min\{(1 - \frac{at}{S_0})v_f + \frac{at}{S_0}v_p, v_f + a_{max}t\}, v_f - b_{max}t\} \\ S' = S_0 + (v_p - v_f)t \end{cases} \quad (3.29)$$

The mobility limitation ensures that following observation can only be adjacent with the previous one, so that we can use "actions" (up, down, or hold) to represent this transition. Meanwhile, lots of redundant options can be ignored this way, since their correspond areas are not adjacent and the mobility limitation guarantees that the probability of this event will be 0. The probabilities P for all combinations on different "actions" (up, down,hold) in different dimensions $\langle v, S \rangle$ can be calculated with the Equation (3.29). Finally, back to the front, we can define the corresponding events and have the probabilities to form the emission matrix.

Denotations used particularly in this section are shown as follow. An observation indicates a two-dimension pair including the velocity and spacing. Event O_i represents that the vehicle locates in the i_{th} observation according to our discretization and decoding procedure, event A_j represents that the vehicle was in the j_{th} observation area a Δt time before, and event B_k represent that its preceding vehicle travel at the speed in the k_{th} interval. The probability $P(O_i)$ would be the elements in the emission matrix. According to the law of total probability, probability $P(O_i)$ can be achieved by sampling from event A_j . Due to the mobility constraints, $P(O_i|A_j) = 0$, where i and j are not adjacent areas. In addition, $P(A_j) = \frac{1}{M}$ since the sampling in original events is even

3. ENHANCED MOBILITY MODEL WITH HMM

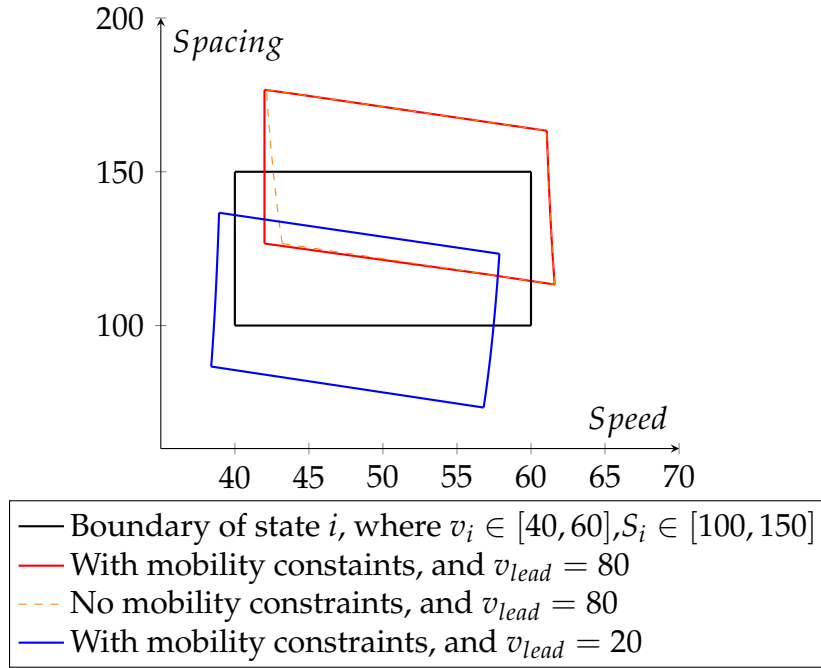


Figure 3.8: Impacts on different speed of preceding vehicle

and there are totally M kinds of observations:

$$\begin{aligned}
 P(O_i) &= \sum_j^M P(O_i|A_j)P(A_j) = \frac{1}{M} \sum_{adjacent_j} P(O_i|A_j) \\
 &= \frac{1}{M} \sum_{adjacent_j} \sum_k P(O_i|A_j, B_k)P(B_k)
 \end{aligned}$$

where $P(O_i|A_j, B_k)$ is the probability in Equation (3.28). Then,

$$P(O_i) = \frac{1}{M \cdot v_{max}} \sum \sum \frac{\int \int_{j,k} dv dS}{\int \int_{\Sigma} dv dS} \quad (3.30)$$

For all O_i , $P(O_i)$ can be calculated respectively. As a consequence, the emission could be achieved by sampling the speed of the preceding vehicle v_p through all the reasonable value and sum up the probability for each observation area. According to this adaptation method, emission matrix of other dynamics model can be derived similarly.

In summary, the adaptation algorithm for 2-dimension can be found as in

3.2 Traditional Mobility Models and their adaptation

	S1	S2	S3	S4	S5
V1	0.0183	0.0429	0.0403	0.0402	0.0608
V2	0.0203	0.0357	0.0330	0.0327	0.0452
V3	0.0314	0.0354	0.0336	0.0333	0.0367
V4	0.0830	0.0632	0.0628	0.0630	0.0411
V5	0.0382	0.0296	0.0301	0.0306	0.0187

Table 3.1: Adaptation result for a demonstration

Algorithm 7. As for an example of the car-following model (Equation (3.24)), if there are 5 intervals on both velocity and spacing, in which case velocity is divided by 20ft./s and spacing is discretized by 50 feet, and we evaluate $k = 1, l = 1, m = 0$, parameter $\alpha = 12, \tau = 2/3$, meanwhile sampling of the input velocity and spacing are evenly, we may have an adaptation result as shown in Table 3.1.

Algorithm 7 Adaptation Method in 2-dimension

Input:

Dynamics representation of a car-following model in 2-dimension: $\mathbf{out} = f(\mathbf{in})$;

Output:

Probability representation of a car-following model. As for a discrete situation, the probability distribution is described by its probability mass function (pmf): $\mathbf{P}(out_i)$.

- 1: Search for a closed loop of *output* variables;
 - 2: Deduct with kinetic model to obtain iterative form of both relevant variables;
 - 3: According to the Total Probability Law, Sample by the *input* variables, until you have the pmf described by the meta probability $\mathbf{P}(out_i|in_1 \cdots in_j)$;
 - 4: A = full area of evaluation zone;
 - 5: **for** Every possible i , that o_i is in sample space **do**
 - 6: A_i = overlapping area of indicated zone and evaluation zone;
 - 7: Meta probability $\mathbf{P}(out_i|in_1 \cdots in_j) = \frac{A_i}{A}$;
 - 8: **end for**
 - 9: **for** Each *input* variables in_j **do**
 - 10: $\mathbf{P}(in_j)$ can be calculated according to the sampling rule of variable in_j ;
 - 11: **end for**
 - 12: $\mathbf{P}(out_i) = \sum_{in_1=1}^{N_1} \cdots \sum_{in_j=1}^{N_j} \mathbf{P}(out_i|\mathbf{in})\mathbf{P}(in_1) \cdots \mathbf{P}(in_j)$.
-

3.2.4 Summary on Adaptation method of Mobility Models

In the first part of this chapter, the research context of mobility model has been explained at first. Basic categories on VMM have been given at the beginning of this chapter. There exist multiple categories that based on different principles. These categorizations are overlapping, which means a certain VMM may multiple kinds of classes of MM from a different point of view.

Then, traditional VMM(MM)s have been summarized and the common presentations on movement restrictions are revealed. These restrictions include the temporal dependency restriction and the spatial dependency restrictions. Both of them are used to characterize certain specific behaviors of a vehicle.

Next, we claim the differences between some probabilistic based VMM and the target of our adaptation: a VMM with its probabilistic representation. These two concepts should be well distinguished. In our opinion, every previous introduced traditional VMM can have its corresponding representation in probability. To be noticed that, traditional here means that it is expressed by dynamics equations; it is used to highlight the difference in representation.

At last, the adaptation method is given in two steps, starting with a basic adaptation in 1-dimension to demonstrate the basic concept of this adaptation. Certain assumptions have been made to simplify the situation and finally, the probability of each possible output can be deduced. In a word, by sampling from the different input, the output frequency is used to represent the final probability. After in 1-dimension, a more practical example in 2-dimension is given by the adaptation of car-following model, which is also being used in the following chapters. 2-dimension situation becomes more complex. However, coming with the same basic concept, the probabilities are still calculable. The sampling method is still open to question: as in this chapter, it is evenly sampled, while there is also possible to iterate this behavior multiple times until a convergent representation is obtained.

With this adaptation method, it is not only about achieving the corresponding probabilistic representation VMM of previous traditional VMM, but also an important improvement that makes the VMM more flexible, because a probability representation fits with data-driven deep learning method much more than the traditional ones. We will see this in the following section.

3.3 HMM for Mobility Model improvements

This obtained mobility model in Chapter 2 performs better in large scale traffic situation, i.e. the average velocity of the whole platoon of vehicles is closer to a real situation than the velocities from model-based mobility model in a customized virtual sudden stop scenario. However, the method also shows some drawbacks that influence the prediction of vehicle in long-term. The simulation would go wild and the credibility of the simulation would decrease due to the accumulation of errors in a certain period of time.

Based on the current method, in order to neutralize the impact of defects and augment the credibility of the learning result, possible solution would be either

- Optimize the data source, different types of traffic trajectories are now gathering within one same database. These trajectories came from different types of vehicles of different characteristic kinds of drivers of diverse traffic situations. The comprehensiveness of the data may have dragged down the specificity of data.
- Deepen and complicate the neural network structure, thus the network would be capable in containing such a comprehensive data. Yet much more resources would have to be casted in training the deeper network.

Solutions of the problem has been discussed as in [Zhang *et al.* \(2018\)](#), but still we can summarize and dig into it one step deeper. Considering about the constraints, the problem is simplified and described as below. Highway traffic trajectory data has collected in a duration of T frames. Each query of data contains the basic information of basic driving condition. These attributes of vehicles, such as speed and position, are observable, they can be denoted by

$$O = \{O^{(1)}, O^{(2)}, \dots, O^{(n)}, \dots\} \quad (3.31)$$

where $O^{(n)}$ indicates the observation of vehicle No. n , and we have:

$$O^{(n)} = \{o^{(n)}(1), o^{(n)}(2), \dots, o^{(n)}(t), \dots, o^{(n)}(T_n)\} \quad (3.32)$$

3. ENHANCED MOBILITY MODEL WITH HMM

in which

$$o^{(n)}(t) = \{v^{(n)}(t), a^{(n)}(t), x^{(n)}(t), y^{(n)}(t), \dots\} \quad (3.33)$$

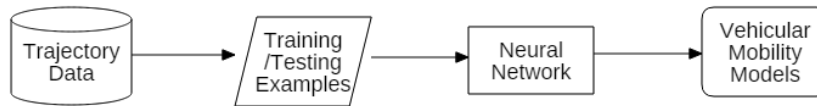
where $v^{(n)}(t)$ is the velocity of the vehicle n at time t , $a^{(n)}(t)$ is the acceleration of the vehicle n at time t , $x^{(n)}(t)$ and $y^{(n)}(t)$ are the coordinate of the vehicle n at time t , T_n indicates the duration of observation on the vehicle n lasts T_n frames.

Whereas, a driving state could be influenced by different issues, such as the psychological state of the driver, driving tendency of different regions, etc.. So the latent driving state are not observable and it will decide the change of these observable attributes in different patterns. Corresponding sequences of driving states can be denoted as $Q = \{Q^{(1)}, Q^{(2)}, \dots, Q^{(n)}\}$. Different driving states lead to different pattern of observable actions. If the transition between states is considered to be probable, the conditions of HMM are met. Then the mobility model can be modeled after an HMM, and the problem can be converted into the three basic problems of HMM.

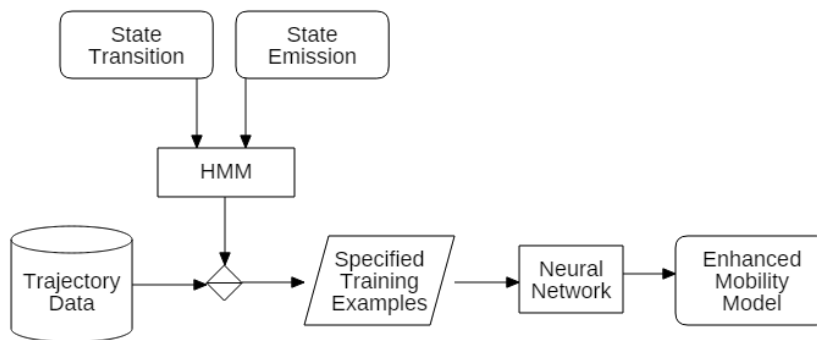
This enhancement can be represented as shown in Figure 3.9. Before sending the data to form demonstration on examples for learning, a specification procedure has been added between the data and neural network. This specification is completed by introducing an HMM. We use HMM to restore the various driving state information during traveling, and the trajectory data can be specified by its estimated driving state. In the enhanced method with HMM, there is a dependency chain as shown in Figure 3.10. The specified data for training and testing examples requires being labeled by an estimated HMM. Moreover, the initialization of this HMM requires the partial knowledge on the pre-defined driving state, which refers to its probability representation on the emit observations. Such a probability distribution of the defined driving model can be achieved by using an adaptation method described in Chapter 3 on the traditional model.

Thus, we aim to develop an enhanced mobility model for traffic simulation, in which the driving states are distinguished according to an HMM, whose parameter can be achieved by solving the learning problem given by the observations from trajectory data O . The distinguishment can be done by solving the decoding problem of HMM given the estimated HMM: $\bar{\lambda}$. Then a more specified mobility model can be built from the neural network based method

3.3 HMM for Mobility Model improvements



(a) Flow chart of the original NN based method



(b) Flow chart of the enhanced NN based method

Figure 3.9: Differences of the flow chart between enhanced method and the original one

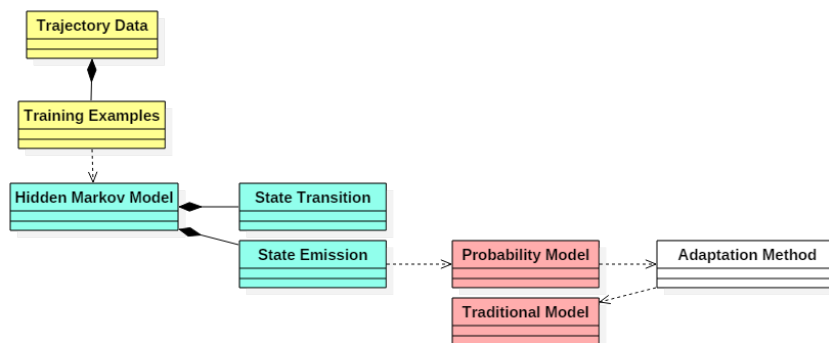


Figure 3.10: Dependency and composition relationship between different components in the enhanced method

3. ENHANCED MOBILITY MODEL WITH HMM

as in Chapter 2 with more precise traffic trajectory data. Finally, the mobility model will be used to reach the corresponding simulation in detail.

3.3.1 HMM in intelligent traffic

Markov property, which indicates that future behavior is decided and only related to the present behavior, is widely used to express the temporal dependency of mobility models. The HMM model has its good performance in nature language processing. It is also widely used to predict the behavior of surrounding vehicles in the field of auto-navigation. In HMM application in traffic systems, the trajectories of vehicles are regarded as the pieces of language, and inferences on the future behavior could be made depending on the current state.

Chen *et al.* (2010) estimated and obtained the headway/spacing transition matrix for the model. Analysis has been done that in different speed conditions, either headway or spacing model would be more proper to be used. Xu *et al.* (2017) imported the HMM into the car-following model, different driving states are given to a traditional Stimulus-Response Model (SRM) to have a better understanding of the behaviors of on road vehicles in auto-navigation.

Comparing with model in (Chen *et al.* (2010)), in which the possible speed range is divided into equal intervals, the establishment of different states in (Xu *et al.* (2017)) seems to be more reasonable that a (Hierarchical Dirichlet Process) HDP-HMM method is used to learn and clustering the most possible different states within the used data. However, the states in both methods seem to be meaningless. Different states learned by HDP have no actual meanings. The represented corresponding actual states are given by analyzing the achieved state that closes to real-world behavior. It is done differently in this chapter.

3.3.2 HMM: Basic knowledge and its denotation

3.3.2.1 Definitions and denotations

Basic denotations of a discrete HMM will be introduced in this section. An HMM λ , which is made up of hidden states denoted as $Q = \{q_i | i \in [1, N]\}$ and discrete observations $O = \{o_k | k \in [1, M]\}$, could be then defined and

3.3 HMM for Mobility Model improvements

represented by a triple tuple $\lambda = \langle A, B, \pi \rangle$, where A stands for the transition matrix among states

$$A = \{a_{ij} | a_{ij} = P(q_t = i, q_{t+1} = j | \lambda), i, j \in [1, N], t \in [1, T - 1]\} \quad (3.34)$$

B stands for the emission matrix

$$B = \{b_j(o_k) | b_j(o_k) = P(o_t = k | q_t = j, \lambda), j \in [1, N], k \in [1, M], t \in [1, T]\} \quad (3.35)$$

and π stands for the initial state possibility distribution

$$\pi = \{\pi_i | \pi_i = P(q_1 = i | \lambda), i \in [1, N]\} \quad (3.36)$$

The coin toss model or the ball and urn model is usually used to explain the meaning of HMM and its parameters. N represents the number of hidden states, which is not directly observable. In the examples, it would be the different weight distribution of coin or the choice of different urns. M represents the number of possible kinds of observations. In the examples, it would be $M = 2$ for coin toss: head and tail or the color of chosen balls from the urn. Emission of a hidden state will be the probability of different observations, such as the probability of head and tail depends on the weight distribution of different coin and the probability of different color depends on the composition of balls in urn.

3.3.2.2 The three problems

Given an HMM as in the previous section, three basic problems interest us if we want to use the model in solving real-world applications. The problems are defined as following:

- Evaluation Problem: given certain observation sequence $O = o_1 o_2 \dots o_T$ and a model $\lambda = \langle A, B, \pi \rangle$, how to efficiently compute the probability of this observations by the model $P(O | \lambda)$?
- Decoding Problem: given certain observation sequence $O = o_1 o_2 \dots o_T$ and the model $\lambda = \langle A, B, \pi \rangle$, how to choose a corresponding state sequence $Q = q_1 q_2 \dots q_T$ that makes sense the most?

3. ENHANCED MOBILITY MODEL WITH HMM

- Estimation Problem: given certain observation sequence $O = o_1 o_2 \dots o_T$, how to adjust the model parameters $\lambda = \langle A, B, \pi \rangle$ to maximize $P(O|\lambda)$?

In the estimation problem, we try to optimize the model parameters to best describe how a given observation sequence comes out. The observation sequence used to adjust the model parameters is called a training sequence, and it is used to train the HMM. So it is also called the training problem and by solving it, it is possible to create best fit models for real phenomena.

In order to solve these problems, first, consider on fixed state sequence $Q = q_1 q_2 \dots q_T$. The probability of the observation sequence O for this state sequence is

$$\begin{aligned} P(O|Q, \lambda) &= \prod_{t=1}^T P(o_t|q_t, \lambda) \\ &= b_{q_1}(o_1) \cdot b_{q_2}(o_2) \dots b_{q_T}(o_T) \end{aligned} \quad (3.37)$$

where we have the assumption on statistical independence of observations. Based on the definition, the probability of such a state sequence Q would be

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \quad (3.38)$$

Then the probability in problem 1 could be calculated as

$$\begin{aligned} P(O|\lambda) &= \sum_Q P(O, Q|\lambda) \\ &= \sum_Q P(O|Q, \lambda) P(Q|\lambda) \\ &= \sum_Q \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T) \end{aligned} \quad (3.39)$$

One problem remains that all possible state sequences should be considered. Thus, there would be too much of calculation, which would be about $2T \cdot N^T$ calculations. It is not called calculable since it is exponential.

So, a forward-backward procedure (Baum (1968); Baum & Eagon (1967)) can be used to solve it. In the forward-backward procedure, a forward variable $\alpha_t(i)$ is defined as

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = i|\lambda) \quad (3.40)$$

which is the probability of the partial observation sequence until time t and

3.3 HMM for Mobility Model improvements

the state i at time t , given the model λ ; and the backward variable $\beta_t(i)$ as

$$\beta_t(i) = P(o_{t+1}o_{t+2} \dots o_T | q_t = i, \lambda) \quad (3.41)$$

which is the probability of the partial observation sequence from time $t + 1$ to the end, given state i at time t and the model λ .

Consider the meaning forward variable and backward variable, it is obvious that by summing up through state q_t , we have

$$\begin{aligned} P(O|\lambda) &= \sum_{i=1}^N P(O, q_t = i | \lambda) \\ &= \sum_{i=1}^N \alpha_T(i) \end{aligned} \quad (3.42)$$

To be noticed, $\alpha_t(i)$ can be initialized as $\alpha_1(i) = \pi_i b_i(o_i)$, and be inducted as

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}) \quad (3.43)$$

in condition of $t \in [1, T - 1]$ and $j \in [1, N]$.

So $\alpha_T(i)$ can be calculated according to this induction by a dynamic programming method (Borne *et al.* (2013)) in much lesser time (in order of N^2T calculations).

Similarly, the induction can be done on $\beta_t(i)$ as it can be initialized as $\beta_T(i) = 1$, and be inducted as

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad (3.44)$$

in condition of $t \in [1, T - 1]$ and $i \in [1, N]$.

To implement a solution to the decoding problem, another variable is defined as

$$\gamma_t(i) = P(q_t = i | O, \lambda) \quad (3.45)$$

which is the probability of being in state i at time t , given the observation sequence O , and the model λ . $\gamma_t(i)$ can be expressed as the forward-backward

3. ENHANCED MOBILITY MODEL WITH HMM

variable, it has the relation with $\alpha_t(i)$ and $\beta_t(i)$ as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (3.46)$$

Using $\gamma_t(i)$, it is able to solve the individually most likely state q_t at time t as

$$q_t = \arg \max_{1 \leq i \leq N} [\gamma_t(i)] \quad (3.47)$$

in condition of $t \in [1, T]$

But, the transition between states is not considered. For example, if $q_t = i$ and $q_{t+1} = j$ happens to be the most likely state by solving Equation (3.47), but it turns out that the transition probability could be $a_{ij} = 0$, which means the sequence is not a valid state sequence.

A formal valid technique to find the best state sequence is called the Viterbi algorithm (Forney (1973); Viterbi (1967)). $\delta_t(i)$ is defined as the highest probability along a single path, at time t , which accounts for the first t observations and ends in state i .

$$\delta_t(i) = \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_{t-1}, q_t = i, o_1 o_2 \dots o_t | \lambda) \quad (3.48)$$

where its induction is as

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(o_{t+1}) \quad (3.49)$$

Pairing with $\delta_t(i)$, $\psi_t(i)$ is used to represent the path that maximizes $\delta_t(i)$. It has induction similarly

$$\psi_{t+1}(j) = \arg \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] \quad (3.50)$$

Then the probability of whole path should be

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (3.51)$$

3.3 HMM for Mobility Model improvements

and path should be the backtracking on $\psi_t(i)$:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t \in [1, T - 1] \quad (3.52)$$

At last, there is the third problem: estimation problem. The estimation problem is to determine a method to adjust the model parameters $\lambda = \langle A, B, \pi \rangle$ to maximize the probability of the observation sequence given the model. There is no analytic solution for the problem. Practically, an expectation maximization like method is used (Or say the Baum-Welch method (Baum & Petrie (1966))). In order to describe the procedure, another variable $\xi_t(i, j)$ shall be defined.

$\xi_t(i, j)$ stands for the probability of being in state i at time t and state j at time $t + 1$ given the model and the observation sequence.

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda) \quad (3.53)$$

According to the definition of forward and backward variables,

$$\begin{aligned} \xi_t(i, j) &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned} \quad (3.54)$$

The estimation can be started with $\gamma_t(i)$ and $\xi_t(i, j)$. If $\gamma_t(i)$ is summed over time, it would be the quantity of expected number of times that state i is visited. If $\xi_t(i, j)$ is summed over time, it is the expected number of transition from state i to j .

With all these variables, the new model $\bar{\lambda}$ can be estimated by:

$$\bar{\pi}_i = \gamma_1(i) \quad (3.55)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (3.56)$$

$$\bar{b}_j(o_k) = \frac{\sum_{t=1, o_t=k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (3.57)$$

3. ENHANCED MOBILITY MODEL WITH HMM

More specifically, estimation process can be achieved by maximizing Baum's auxiliary function

$$Q(\lambda, \bar{\lambda}) = \sum_Q P(Q|O, \lambda) \log[P(O, Q|\bar{\lambda})] \quad (3.58)$$

It has been proven that maximization of $Q(\lambda, \bar{\lambda})$ leads to increased likelihood, so that the estimation would be implemented as

- E (expectation) step: calculation of the auxiliary function $Q(\lambda, \bar{\lambda})$
- M (maximization) step: maximization over $\bar{\lambda}$

Iteration will stop when it meets the pre-defined stop condition, by which the third problem is solved.

As we want to maximize the auxiliary function Equation (3.58), it can be adapted according to the equation below

$$P(Q|O, \bar{\lambda}) = \frac{P(Q, O|\bar{\lambda})}{P(O|\bar{\lambda})} \quad (3.59)$$

$P(O|\bar{\lambda})$ is considered a constant as for a certain model $\bar{\lambda}$. it does not influence the maximization of $Q(\lambda, \bar{\lambda})$ over $\bar{\lambda}$. So, it is equal to maximize the function below than to maximize $Q(\lambda, \bar{\lambda})$:

$$Q(\lambda, \bar{\lambda}) = \sum_Q P(Q, O|\bar{\lambda}) \log P(Q, O|\lambda) \quad (3.60)$$

Applying the expansion in Equation (3.39), we have

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum_Q P(Q, O|\bar{\lambda}) \log P(Q, O|\lambda) \\ &= \sum_Q P(Q, O|\bar{\lambda}) \log \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T) \\ &= \sum_Q P(Q, O|\bar{\lambda}) \log \pi_{q_1} + \sum_Q P(Q, O|\bar{\lambda}) \left(\sum_{t=1}^{T-1} \log a_{q_t q_{t+1}} \right) \\ &\quad + \sum_Q P(Q, O|\bar{\lambda}) \left(\sum_{t=1}^T \log b_{q_t}(o_t) \right) \end{aligned} \quad (3.61)$$

By discussing the possibilities of all state sequence Q , we have

$$\begin{aligned}
 Q(\lambda, \bar{\lambda}) &= \sum_{i=1}^N P(q_1 = i, O|\bar{\lambda}) \log \pi_i \\
 &+ \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^{T-1} P(q_t = i, q_{t+1} = j, O|\bar{\lambda}) \log a_{ij} \\
 &+ \sum_{j=1}^N \sum_{t=1}^T P(q_t = j, O|\bar{\lambda}) \log b_j(o_t)
 \end{aligned} \tag{3.62}$$

Setting up a standard Lagrange optimization using Lagrange multipliers, $Q(\lambda, \bar{\lambda})$ is maximized when the three parts of it are maximized, where there are basically the same expressions as in Equation (3.55), (3.56), (3.57).

- Based on the condition of $\sum_{i=1}^N \pi_i = 1$:

$$\bar{\pi}_i = \frac{P(q_1 = i, O|\bar{\lambda})}{P(O|\bar{\lambda})} \tag{3.63}$$

- Based on the condition of $\sum_{j=1}^N a_{ij} = 1$:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} P(q_t = i, q_{t+1} = j, O|\bar{\lambda})}{\sum_{t=1}^{T-1} P(q_t = i, O|\bar{\lambda})} \tag{3.64}$$

- Based on the condition of $\sum_{k=1}^M b_j(o_t = k) = 1$:

$$\bar{b}_j(o_t = k) = \frac{\sum_{t=1}^T P(q_t = i, o_t = k|\bar{\lambda})}{\sum_{t=1}^T P(q_t = i, O|\bar{\lambda})} \tag{3.65}$$

In summary, all the three problems are solved. The solutions of these three problems can be applied to estimate an HMM by trajectory data, and label the data with corresponding state for NN learning.

3.3.3 HMM combined Mobility Model

Since that the states of HMM are latent and not observable, HDP-HMM is commonly used to determine the states, where they use HDP to cluster and obtain the states by selecting the classes with top scoring and appearance. It

3. ENHANCED MOBILITY MODEL WITH HMM

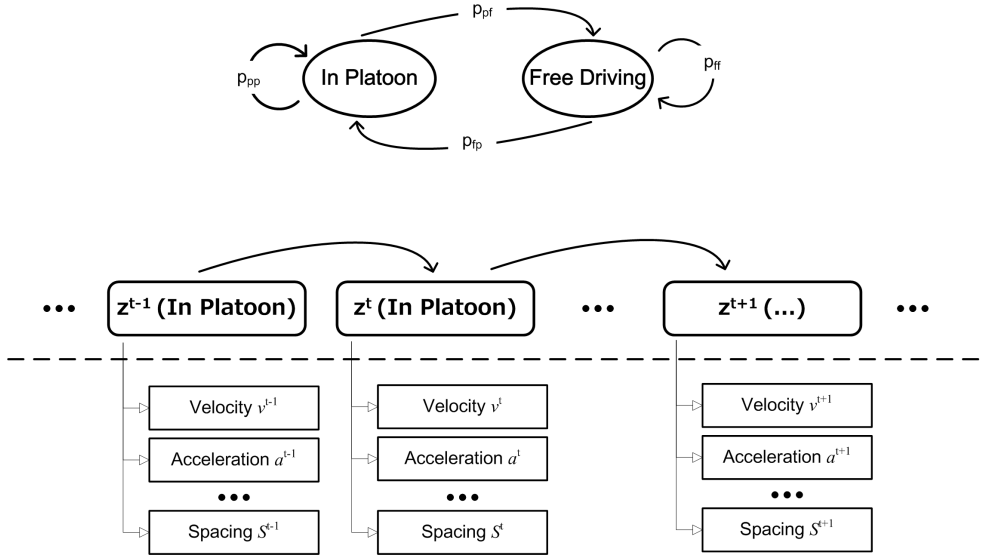


Figure 3.11: Transition and Emission of our HMM

is rough to say that the states from these HDP-HMM are totally meaningless, however the meaning of the states come out after the generation of the states. By analyzing and examining the performance attributes, states are defined.

While in our opinion, the states can be pre-defined. For instance, a vehicle would travel either at an in-platoon state or at a free-driving state. Vehicle within an in-platoon state must possess certain characteristics, for example, while traveling at in-platoon state, vehicle behaves "normally" with traffic rules fully respected and no extreme or dangerous actions taken, hypothesis of corresponding platoon dynamics based method are met and it can be used to describe the situation. Other behaviors can be broadly defined as free driving. While traveling at free driving state the vehicle is out of platoon and seems less predictable. Regular model-based method meets its limitation here, thus a learning-based method as described in Chapter 2 is employed to solve this dilemma (Zhang & El Kamel (2018)). Labeling the original data by solving a decoding problem of HMM, the corresponding period of trajectories are filtered out, then the learning is more efficient and more effective. This division is both reasonable and executable. These two states are both mutually exclusive and complementary.

More specifically, in this section, $N = 2$ (in-platoon state and free-driving state as described above). As a consequence, the transition matrix will be in

3.3 HMM for Mobility Model improvements

scale of 2×2 . It should also be noticed that the dimension N is not strictly limited to 2 for all HMMs. It could be any number that is reasonable to the corresponding problem. Since the states are defined as above, in-platoon state and free-driving state, $N = 2$ is the solution of our problem, as all other behavior else than in-platoon state can be regarded as a behavior of free-driving state.

The emissions have not been decided yet, it should be made up of observable attributes of the vehicles that must also exist in the data collections. In order to simplify the situation and avoid redundant calculation, weakly relevant attributes are neglected, such as *veh_type* for vehicle's type, *veh_length* for vehicle's length, and any global coordinates x_{global}, y_{global} . Referring to a selected traditional model, the Gipps's car-following model will demonstrate our in-platoon driving mode. As described in its expression, speed and spacing (distance to the preceding vehicle) are the emission on observable attributes of a vehicle. Then the remaining problem would be how do we treat these two attributes.

If they are independent to each other, then two independent HMM can be built, whose emission can be obtained using the method described as in Chapter 3.3. However, if there is dependency between the two attributes, then the method described as in Chapter 3.4 will fit better.

In order to find out an answer of this problem, the marginal distribution and joint distribution of these two attributes are examined. Results can be found as in Figure 3.12, while corresponding data is given by Table.3.2. Values in this table are the count of appearances frequency of corresponding events in the database. The row sum stands for the marginal distribution of spacing, and the col sum represents the marginal distribution of speed. The frequency can be seen as the probability of corresponding events, and to be noticed, the total sum equals to 1 as shown in the table.

In Table 3.3, one thing should be noticed that frequencies counted from original traffic data are considered to represent the traffic system. It demonstrates the deviation between product of single frequencies/probabilities and joint frequencies/probabilities. The Δ is expressed by the derivation on percentage to $P(\mathbf{A}, \mathbf{B})$, if there is no observation of joint event (A, B) , Δ is denoted as 100%. If event \mathbf{A} and \mathbf{B} are conditional independent, then the derivation Δ

3. ENHANCED MOBILITY MODEL WITH HMM

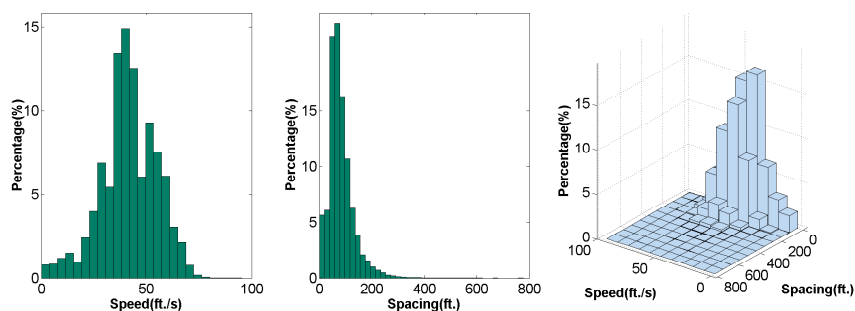


Figure 3.12: Histogram of marginal and joint distribution: (Left) Marginal distribution of Speed; (Middle) Marginal distribution of Spacing; (Right) Joint distribution of Speed and Spacing

Table 3.2: Joint and marginal distribution of Speed and Spacing

$P(A, B)$	[0,20) A_1	[20,40) A_2	[40,60) A_3	[60,80) A_4	[80, $+\infty$) A_5	row sum
[0,100) B_1	0.0585	0.3288	0.3045	0.0512	0.0000	0.7431
[100,200) B_2	0.0005	0.0508	0.1498	0.0278	0.0000	0.2289
[200,300) B_3	0.0001	0.0020	0.0170	0.0045	0.0000	0.0235
[300,400) B_4	0.0000	0.0005	0.0019	0.0006	0.0000	0.0030
[400, $+\infty$) B_5	0.0000	0.0003	0.0010	0.0002	0.0000	0.0015
col sum	0.0591	0.3823	0.4743	0.0843	0.0000	Sum = 1

3.3 HMM for Mobility Model improvements

Table 3.3: Deviation on frequency/probability of corresponding events

Event Code	$P(A) \cdot P(B)$	$P(A, B)$	Δ
A_1, B_1	0.0439	0.0585	24.9%
A_1, B_2	0.2839	0.3288	13.7%
A_1, B_3	0.3522	0.3045	15.7%
A_1, B_4	0.0626	0.0512	22.3%
A_1, B_5	0.0004	0.0000	100%
A_2, B_1	0.0135	0.0005	2600%
A_2, B_2	0.0874	0.0508	72.2%
A_2, B_3	0.1085	0.1498	27.6%
A_2, B_4	0.0193	0.0278	30.6%
A_2, B_5	0.0001	0.0000	100%
A_3, B_1	0.0014	0.0001	1300%
A_3, B_2	0.0090	0.0020	350%
A_3, B_3	0.0112	0.0170	34.1%
A_3, B_4	0.0020	0.0045	55.6%
A_3, B_5	0.0000	0.0000	0%
A_4, B_1	0.0002	0.0000	100%
A_4, B_2	0.0012	0.0005	140%
A_4, B_3	0.0014	0.0019	26.3%
A_4, B_4	0.0003	0.0006	50.0%
A_4, B_5	0.0000	0.0000	100%
A_5, B_1	0.0001	0.0000	100%
A_5, B_2	0.0006	0.0003	100%
A_5, B_3	0.0007	0.0010	30.0%
A_5, B_4	0.0001	0.0002	50.0%
A_5, B_5	0.0000	0.0000	0%

3. ENHANCED MOBILITY MODEL WITH HMM

in the table should be close to zero. According to the joint probability equation, variable X and Y is independent if it meets

$$P(X = x, Y = y) = P(X = x) \cdot P(Y = y) \quad (3.66)$$

As can be seen from the listed data, it can be conclude that events on the attributes *Speed* and *Spacing* are not probabilistic independent, so that it is not possible to consider them separately as two individual HMMs. Yet since the meaning of emission in HMM does not stand for any actual influences on the model, joint attributes can be encoded by a discretization procedure. Consequently, the dimension of emission is reduced.

$$obv = f[v_i^{(n)}(t), \Delta x_i^{(n)}(t)] \quad (3.67)$$

The basic modeling of the HMM for our mobility model has been decided, including the dimensions on state: N and format of emissions: *obv*. In the following sections, several issues that we encountered in implementation will be discussed .

3.3.3.1 Scaling Problem

One thing should be paid attention in practice is the scaling problem ([Shen \(2008\)](#)).

Since α and β are probabilistic variables, which evaluate within interval $[0, 1]$. Their computing would loose its accuracy when the length of sequence grows large, even in double precision variables. Solution to the accuracy problem is a scaling procedure that refers to [Rabiner \(1989\)](#).

A scaling coefficient will solve the problem. In the computation of forward and backward variable α and β , a scaling coefficient c is defined as

$$c_t = \frac{1}{\sum_{i=1}^N \check{\alpha}_t(i)} \quad (3.68)$$

where $\check{\alpha}_t(i) = \sum_{j=1}^N \hat{\alpha}_{t-1}(j) a_{ji} b_i(o_t)$, and $\hat{\alpha}_t(i) = c_t \check{\alpha}_t(i)$ for each time step t . Obviously, we have

$$\hat{\alpha}_2(i) = c_2 c_1 \alpha_2(i) \quad (3.69)$$

By conduction

$$\hat{\alpha}_t(i) = c_t \sum_{j=1}^N \hat{\alpha}_{t-1}(j) a_{ji} b_i(o_t) = c_t c_{t-1} \cdots c_1 \alpha_t(i) \quad (3.70)$$

Let $\mathbf{C}_t = \prod_{\tau=1}^t c_\tau$, specially, we have $P(O|\lambda) = 1/\mathbf{C}_T$. Then, the scaled $\alpha_t(i)$ is as

$$\hat{\alpha}_t(i) = \left(\prod_{\tau=1}^t c_\tau \right) \alpha_t(i) = \mathbf{C}_t \alpha_t(i) \quad (3.71)$$

Similarly, the scaled $\beta_t(i)$ as

$$\hat{\beta}_t(i) = \left(\prod_{s=t}^T c_s \right) \beta_t(i) = \mathbf{D}_t \beta_t(i) \quad (3.72)$$

where $\mathbf{D}_t = \prod_{s=t}^T c_s$. We have $\mathbf{C}_t \cdot \mathbf{D}_{t+1} = \mathbf{C}_T$ and $\mathbf{C}_t \cdot \mathbf{D}_t = \prod_{\tau=1}^t c_\tau \cdot \prod_{\tau=t}^T c_\tau = \mathbf{C}_T \cdot c_t$. Then, Equation (3.56) turns to be

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(o_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \hat{\alpha}_t(i) \hat{\beta}_t(j) / c_t} \quad (3.73)$$

Equation (3.57) turns to be

$$\bar{b}_j(o_k) = \frac{\sum_{t=1, o_t=k}^T \hat{\alpha}_t(j) \hat{\beta}_t(j) / c_t}{\sum_{t=1}^T \hat{\alpha}_t(j) \hat{\beta}_t(j) / c_t} \quad (3.74)$$

3.3.3.2 Multi-observation sequences

The estimation of HMM can be done according to Equation (3.55), (3.56), (3.57). But there is a minor difference while applying an estimation in practice.

As well known, our data used to train this HMM is also from NGSIM database. Problem locates at the way how we treat the data and put it into the HMM. There are totally about 3,000 pieces of continuous record in the database, which means we have 3,000 sequences that can be used to learn the model. In order to make full use of all the data becomes a problem, since it is not proper to stitching them into one single sequence. Besides, it is not certain if our model is a left-right model, which is a kind of model that converge into one certain state.

3. ENHANCED MOBILITY MODEL WITH HMM

We have all the observation as

$$O = [O^{(1)}, O^{(2)}, \dots, O^{(k)}] \quad (3.75)$$

where each $O^{(i)} = [o_1^{(i)} o_2^{(i)} \dots o_{T_i}^{(i)}]$ is the i th observation sequence. P_k is denoted as the probability of observation sequence $O^{(k)}$ given the model λ . Then

$$P(O|\lambda) = \prod_{k=1}^K P(O^{(k)}|\lambda) = \prod_{k=1}^K P_k \quad (3.76)$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^k(j) a_{ij} b_j(o_{t+1}^k) \hat{\beta}_{t+1}^k(j)}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^k(i) \hat{\beta}_t^k(i) / c_t^k} \quad (3.77)$$

$$\bar{b}_j(o_k) = \frac{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1, o_t=k}^{T_k} \hat{\alpha}_t^k(j) \hat{\beta}_t^k(j) / c_t^k}{\sum_{k=1}^K \frac{1}{P_k} \sum_{t=1}^{T_k} \hat{\alpha}_t^k(j) \hat{\beta}_t^k(j) / c_t^k} \quad (3.78)$$

3.3.3.3 Partial knowledge estimation

Differs from a traditional HMM, we have partial knowledge of the model in our case: the in-platoon state is specified to a pre-defined dynamics model, which means the emission matrix of this state is designated. It is helpful for us to estimate the model. Because a part of the whole emission matrix is "given", our model can be initialized closer to the "best" solution, and we shall neither need nor be capable to estimate the already given part of the emission matrix. The given information comes our definition on in-platoon driving state, and its corresponding emission can be achieve by applying the adaptation algorithm as in Algorithm 7.

With a slight modification in Equation (3.78), the final estimation is given by Equation (3.79), (3.80). We have \bar{a}_{ij} and $\bar{b}_j(o_k)$:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \hat{\alpha}_t^k(j) a_{ij} b_j(o_{t+1}^k) \hat{\beta}_{t+1}^k(j)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \hat{\alpha}_t^k(i) \hat{\beta}_t^k(i) / c_t^k} \quad (3.79)$$

$$\bar{b}_j(o_k) = \frac{\sum_{k=1}^K \sum_{t=1, o_t=k}^{T_k} \hat{\alpha}_t^k(j) \hat{\beta}_t^k(j) / c_t^k}{\sum_{k=1}^K \sum_{t=1}^{T_k} \hat{\alpha}_t^k(j) \hat{\beta}_t^k(j) / c_t^k}, j \neq i \quad (3.80)$$

In Equation (3.80), the $j \neq i$ constraint indicates that the knowledge of state i has already known, as in Table 3.1. Another thing we should notice is that Baum-Welch algorithm, or Expectation Maximization (EM) algorithm, only gives local maximum solution, and it could stuck at a local maximum point. Thus, the initialization before estimation becomes even more important. The place where you initialize the model decides which local maximum you will reach. Fortunately, since we have already known partial knowledge of the under estimated HMM, our initialization will be near the global maximum.

So far, the estimation of HMM is practically done. An estimated transition and emission matrix can describe the model. After that, a decoding, also known as labeling problem can be solved given by the observation sequence O and the estimated model $\bar{\lambda}$. The most probable state sequence correspond to the observation can be obtained by maximizing the probability $P(Q|O, \bar{\lambda})$. Viterbi algorithm that can be found as in Viterbi (1967) is the most commonly used method to solve the problem. With that, a most probable state for each query in the original database can be labeled. Regardless of the queries in in-platoon states, we may focus on the free driving queries and use them to train a more pointed mobility model as in Zhang & El Kamel (2018).

3.3.4 Performance and result

Data collections of NGSIM take records on the vehicles action during a section of highway with seven lanes. The collection contains 15 minutes, 9,809 frames and 1,055,801 queries (each query describes single vehicle's basic information and its location and velocity in single frame) of recorded traffic data. Quantity of vehicles rises in the first 1000 frames and decreases in the last 1000 frames. It remains stable in the middle of the observation, thus we shall use the data from middle period of the observation.

Environment of our experiments is as following, The experiments have been setup and run with the processor of 2.50GHz and 8.0GB memory. Major experiment platform includes Matlab as for basic data treatment and calculation, Tensorflow Abadi *et al.* (2016) as for the neural network establishment and training and SUMO simulator Krajzewicz *et al.* (2012) as for an intuitive animation traffic simulation output demonstration.

3. ENHANCED MOBILITY MODEL WITH HMM

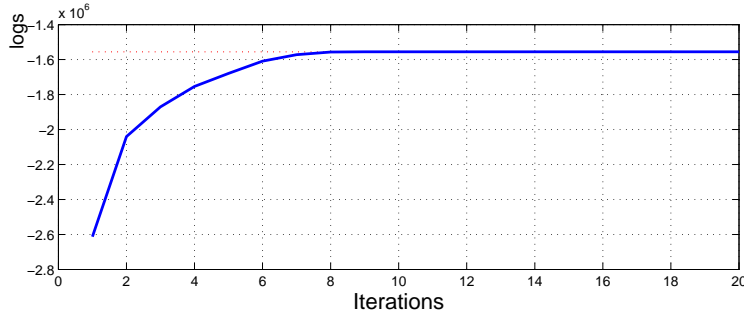


Figure 3.13: Convergence of Baum Welch algorithm for multi-observation sequences, measured by Logarithm-Probability

Table 3.4: Estimated Transition Matrix

	In-Platoon	Free Driving
In-Platoon	0.9707	0.0293
Free Driving	0.0053	0.9947

At the first part of our experiments, we implemented the adaptation from the dynamics model as described in Equation (3.24) to a probabilistic model. $\alpha = 12$ and reaction time $t = \frac{2}{3}$ is chosen. Different α leads to different sensitivity on the derivation of velocity and spacing. 12 is the historical value for most practice. As a result, the emission for state "in-platoon" driving can be found as shown in Table 3.5.

Then, the HMM is estimated. Initialization is very important to an HMM estimation. However, we will not discuss this deeply in this chapter. Taking advantage of the partial knowledge of the model, the initialization of HMM can be done by setup partial of the emission with definite values. As a consequence, the emission for "in-platoon" driving will not change during the iteration of estimation. Logarithm-Probability is used after scaling to avoid "under-flow" problem, as shown in Figure 3.13. As a result, the parameter (transition A and emission B) of HMM is as shown in Table 3.4, 3.5:

Then, the model $\lambda = \langle \bar{A}, \bar{B} \rangle$ is used to label out the states in basic NGSIM database. The decoding procedure is done, and 123,563 frames of data have been labeled as "in-platoon" driving, while 627,593 frames of data have been labeled as "Free" driving. Ignoring the queries that have been labeled as "in-platoon" driving, the other queries are sent to train the neural network. For

3.3 HMM for Mobility Model improvements

Table 3.5: Estimated Emission Matrix

Encoded Emission	1	2	3	4	5	6	
In-Platoon	0.0183	0.0429	0.0403	0.0402	0.0608	0.0203	
Free Driving	0.0535	0.0001	0.0000	0.0000	0.0000	0.1150	
Encoded Emission	7	8	9	10	11	12	
In-Platoon	0.0357	0.0330	0.0327	0.0452	0.0314	0.0354	
Free Driving	0.2809	0.0524	0.0000	0.0000	0.0690	0.2932	
Encoded Emission	13	14	15	16	17	18	
In-Platoon	0.0336	0.0333	0.0367	0.0830	0.0632	0.0628	
Free Driving	0.1359	0.0000	0.0000	0.0000	0.0001	0.0000	
Encoded Emission	19	20	21	22	23	24	25
In-Platoon	0.0630	0.0411	0.0382	0.0296	0.0301	0.0306	0.0187
Free Driving	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

instance, the structure of neural network in this chapter is given by a Feed Forward Neural Network (FFNN) with input at 10 and output at 3 ($v, a, lane$). Learning rate η is chosen at 0.06. Since there is no absolute winner of the learning algorithms, it has been trained by both an LM-algorithm and a BFGS-algorithm. Both results have been shown as in Figure 3.14 and Figure 3.15. The vehicle velocity and its occupied lane have been chosen to demonstrate the simulation result. It can be concluded that although either of them may have an advantage in a specific parameters' setup, the overall result is quite charming, despite there are some abnormal estimations in the occupied lane.

Finally, using the obtained neural network, we can start the simulation with SUMO simulator. The scenario file are generated before, which is a period of highway traffic with 6 normal driving lanes, speed limitation varies and list from high to low from left to right just as real world. An addition lane is located at the side of the six lanes, which is an entrance to the highway in real world. Since the NGSIM database is used to analyze a congestion's forming and resolving, the average speed of both the real world and the simulation would not be high. A perturbation test for the neural network training could be found in Zhang & El Kamel (2018).

3. ENHANCED MOBILITY MODEL WITH HMM

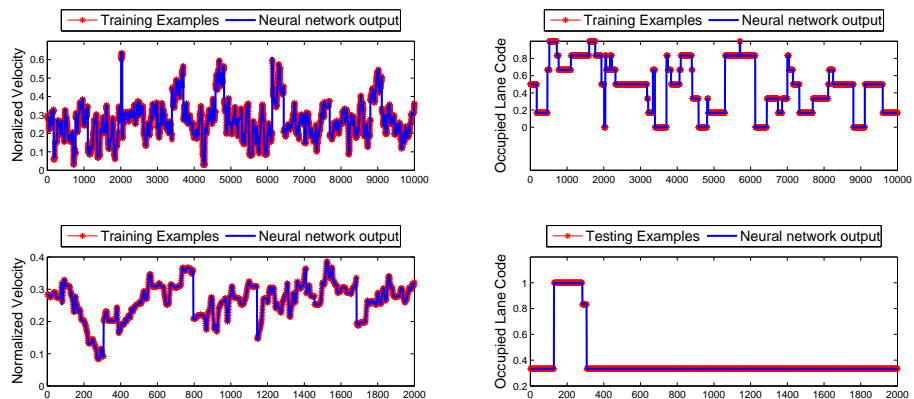


Figure 3.14: Result of Learned NN after HMM labeling by LM-algorithm

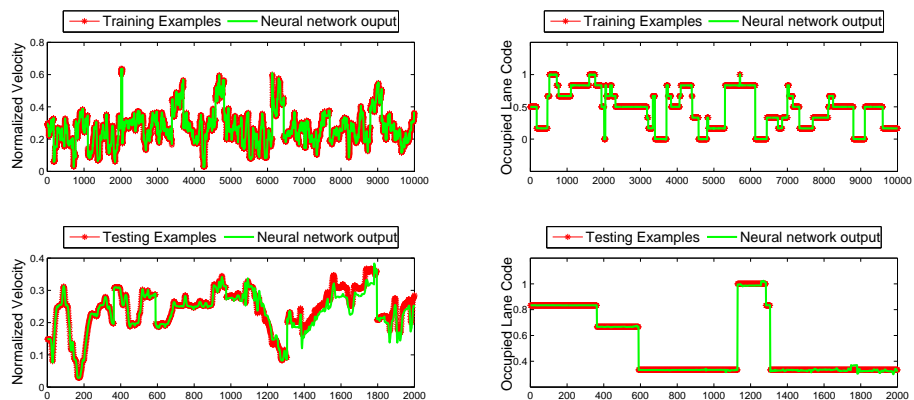


Figure 3.15: Result of Learned NN after HMM labeling by BFGS-algorithm

3.4 Discussion and Conclusion

3.4.1 Discussion

Attention should be paid especially that although our partial knowledge helps us to initialize the Baum-Welch algorithm, performance of the then obtained model after estimation is not guaranteed if the situation is not simplified, i.e. more hidden states exist. As a consequence, the initialization will play an even more important role. Moreover, the observation is discretized in this chapter. A continuous observation HMM can also make sense if combining a Mixed Gaussian Model to characterize the observation. However, there are no critical differences, since this chapter tends more to demonstrate the idea that combining HMM to optimize our previous neural network based method. However, the performance of a simulation is still difficult to measure, since there are no criteria on authenticity.

3.4.2 Conclusion

In this chapter, our previous work on mobility model for virtual traffic simulation is extended. An HMM is introduced to represent the temporal dependency of a driving behavior, and the problem on improving the previously proposed mobility model is converted to solving the three basic problems of an HMM with certain configurations. By digging into the meaning of definitions on different driving states in related works, states of HMM are assigned: one of them is the in-platoon car-following. This is the major difference in the usage of HMM between our method and other HMM related methods, that a hidden state with a definite meaning of driving pattern is pre-defined. Instead, the common usages on HMM tend to assign a meaning to the state derived by clustering.

Corresponding HMM is estimated with the partial knowledge of the pre-defined car-following state. In order to achieve the partial knowledge, an adaptation method from a dynamics model to a probabilistic represented model is used to achieve the corresponding emission of traditional mobility models. Finally, specified data is fed into neural network to obtain final mobility model for simulation. Specification on data is accomplished by solving the decoding

3. ENHANCED MOBILITY MODEL WITH HMM

problem of HMM on the observation in the original database with estimated HMM. Since data used in neural network training is more precise, the performance of mobility model will increase and the credibility of our simulation will also increase. This model is a good tool for simulation.

Chapter 4

Experiment platform and scenario simulation

Contents

4.1	Introduction	101
4.2	UML design of NN-VMM	102
4.2.1	Commercial usage	105
4.2.2	Open-source platform	106
4.3	Scenario simulation	108
4.3.1	Highway scenario reproduction	108
4.3.2	Scenario extension: MIXED NN-VMM/TP-AIM	110
4.3.3	Applications in Transportation technology	116
4.4	Discussion and Conclusion	118

4.1 Introduction

In previous chapters, we've proposed a NN based data-driven mobility model (as in Chapter 2), and improved its performance step by step, by adapting a dynamics based MM into a probabilistic representation one (as in Chapter 3.2) and by integrating the mobility model with an HMM (as in Chapter 3.3) so that different driving state can be represented. In this chapter, an overview will be given on the experiments constructed through all the former chapters and

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

the whole structure of the simulation system used is illustrated, which is denoted as NN-VMM (Neural network based Vehicular Mobility Model). UML diagrams will help us to explain the design of the system. After that, certain simulations are constructed, including a simulation under scenario described as in previous work of my colleague (Liu (2016)). This simulation will demonstrate the applicability of our proposed mobility model, thus future works can be built upon it.

4.2 UML design of NN-VMM

Comparing with the basic system as described in Figure 2.4, the system becomes more complicated due to the addition of HMM and the modification on VMM. As can be seen from Figure 4.1, the NN-VMM is consist of two major parts: the VMM learning part and the Simulation presentation part. To be more precise, the learning part is made up of the neural network learning part and the HMM estimation part. Using the unified modeling language (UML), the relation between these part can be intuitively represented in the design (Luo *et al.* (2011)). Static features of the system are basically the same as in Figure 2.11. Since it has been introduced in previous chapter, we will focus on the dynamic features in this chapter, which can be seen as from corresponding dynamic diagram of UML.

The sequence diagram of UML shows the interactions between objects and components of the whole system (as in Figure 4.2). There are two major interaction groups in this sequence diagram, which describes the procedure of offline learning and online operating of the system respectively.

In the interaction of offline learning, the two messages (requirement on estimation on parameters of HMM and mobility model learning by NN) are independently emitted. According to the restored trajectory sequence data, an estimated HMM can be achieved by solving the corresponding problems as described in Chapter 3.3. It demands an emission description of previous traditional VMM so that training of HMM will not start until it receives the reply from the adaptation. After estimation, the labeling process can begin so that data from NGSIM database will be classified into different categories, and neural network learning can then commence based on different categories. As a

4.2 UML design of NN-VMM

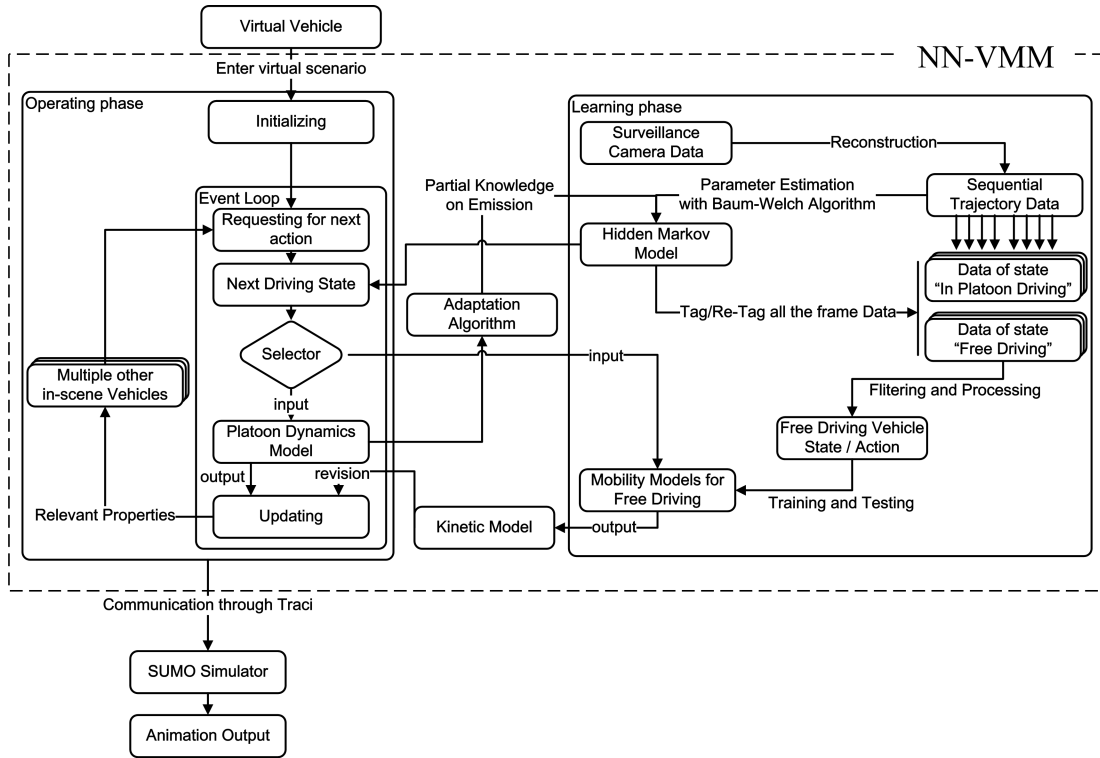


Figure 4.1: Flow Chart of the simulation system NN-VMM

symbol of the accomplishment of learning, this interaction ends. In this off-line learning interaction, HMM and neural network are involved as the two major objects. Interaction is established surrounding these two objects. As a consequence, the enhanced VMM is acquired, which is a necessity in the other interaction.

The second interaction group describes the online operating of the system. enhanced VMM is required when vehicles are assigned to driving with it. Multiple communications and messages are emitted and targeted to the vehicles, which overlaps as shown in the sequence diagram. Thus, another state chart of vehicles is drawn as shown in Figure 4.3. Scenario information conveys the special requirement of the simulation, such as in the scenario simulation as in Chapter 4.3.2, the vehicle can behave as the TP-AIM described in the indicated scenario. In term of a vehicle object, its life circle in the system starts when it is initialized and enters the scenario. At first, it waits for the assignment of driving modes. It can either drive under the enhanced VMM or under certain specified constraints. While driving under control enhanced VMM, detailed

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

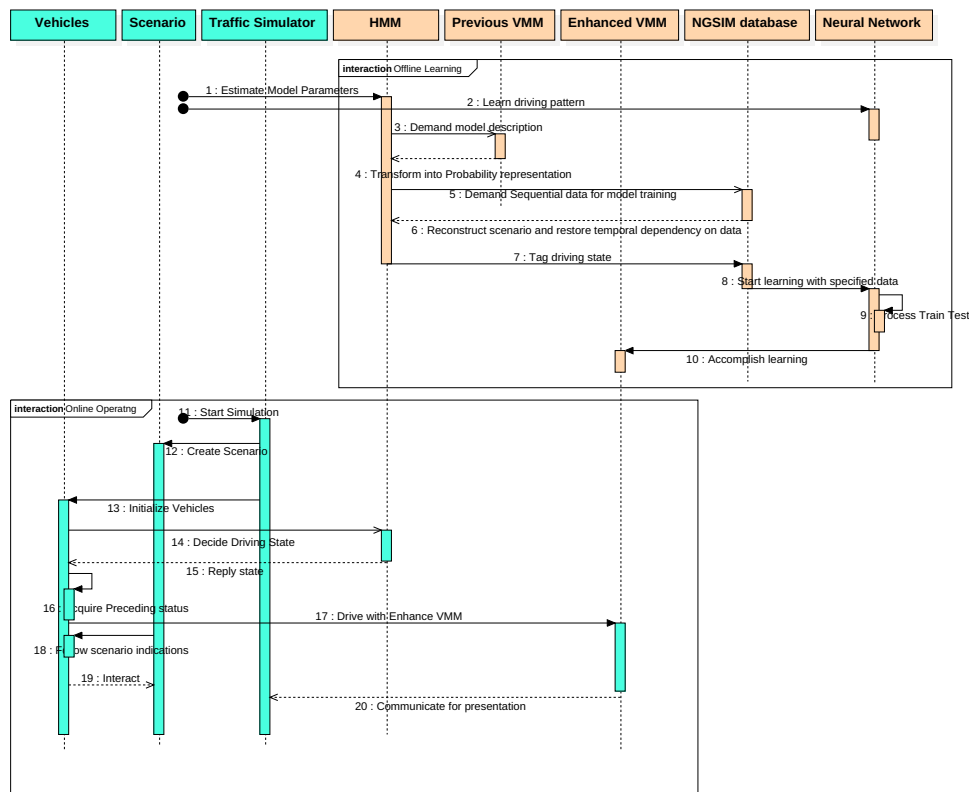


Figure 4.2: Sequence Diagram of the simulation system

VMM is chosen by HMM and different drive mode is employed when states transition happens. Its life circle ends when the scenario is complete.

To be noticed that, each of these parts can be implemented independently, because the off-line learning mode is employed. Although there exists reliability of the result from different parts, since it can be done off-line and no real time requirement has been claimed, the detail implementation of these parts can be replaced by other same functional components. In the following sections, a brief introduction on all the possible software pack used in this system will be given.

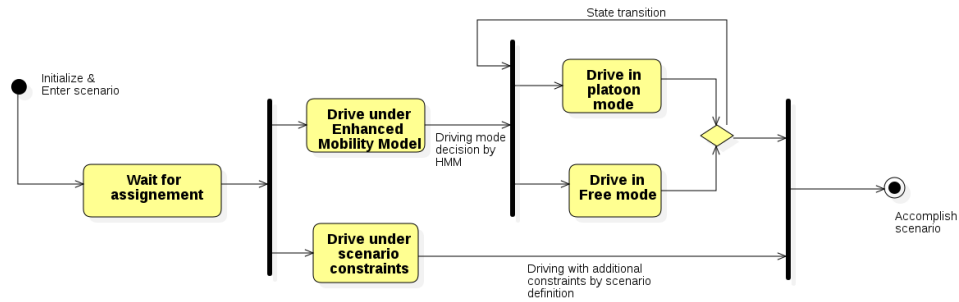


Figure 4.3: State chart Diagram of a vehicle object

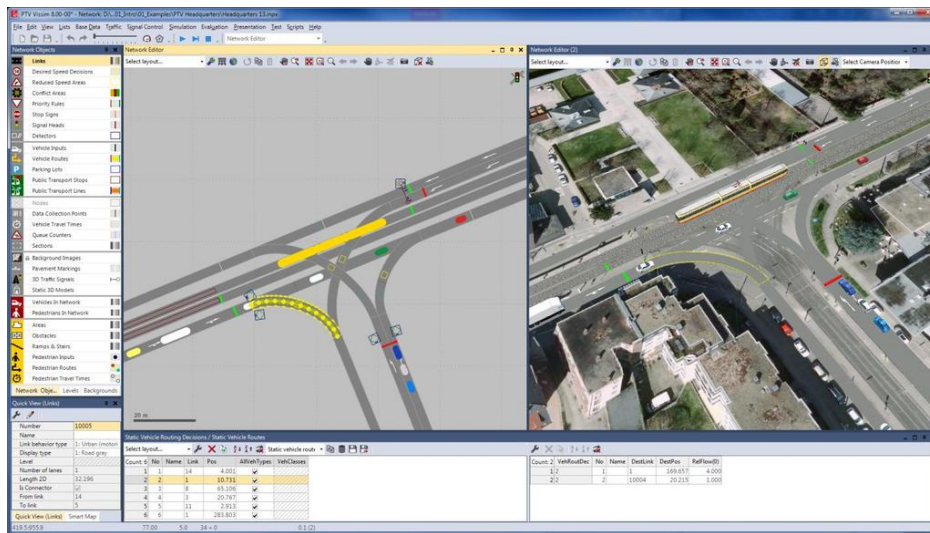


Figure 4.4: User interface of PTV Studio:PTV Vissim

4.2.1 Commercial usage

PTV Studio (Mahmud & Town (2016)) as shown in Figure 4.4 is a comprehensive traffic simulation software including multiple sub-software, such as PTV Vissim for microscopic traffic simulation, which can be used to analyze traffic flow for infrastructure and analyze traffic control or traffic management variants; PTV Visum for macroscopic traffic simulation, which can be used to examine transport network and demands in global level. Recently, a new brand of PTV Viswalk is provided, which enables the user to simulate and model pedestrian walking behaviors. Thus, a more comprehensive and more sophisticated transportation system can be built.

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

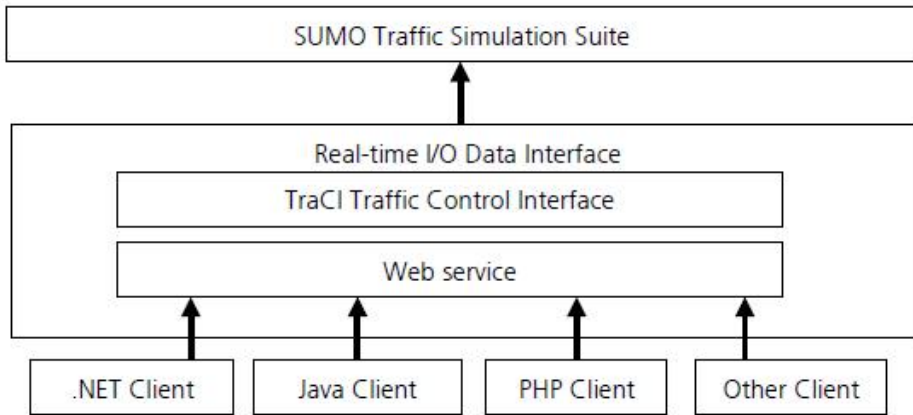


Figure 4.5: Interface TraCI and its relation with SUMO

4.2.2 Open-source platform

SUMO is short for the simulator of urban mobility ([Krajzewicz et al. \(2012\)](#)). It is an open source traffic simulation package known with its high portability. SUMO leaves an interface for programming so that user can customize majority content within a transportation system. The interface is implemented known as TraCI interface ([TraCI \(2018\)](#)), which provides support for major programming languages, including Matlab, C, Python, etc. In characterizing the transportation infrastructure, SUMO employs the node to edge/link to the net structure to describe local traffic which is basically the same as in PTV Studios. A friendly graphic user interface is also provided in SUMO, users can have an intuitive animation result of their simulation.

In the learning part, the Tensorflow ([Abadi et al. \(2016\)](#)) platform from Google is employed for deep learning neural network. Tensorflow is an open-source software library for dataflow programming such as machine learning applications in neural networks. The GPU support of tensorflow calculation enables an acceleration on computation. Yet, its pair tool tensorboard gives the users an intuitive control when testing a neural network. The library also a strong connection with the commonly used numerical processing library so that the user can conveniently deal with their data and import it into the network whatever ways they want.

Speaking of data, the database of NGSIM data ([NGSIM \(2017\)](#)) is used in this thesis. This data collection provides highway trajectory data for vehicle

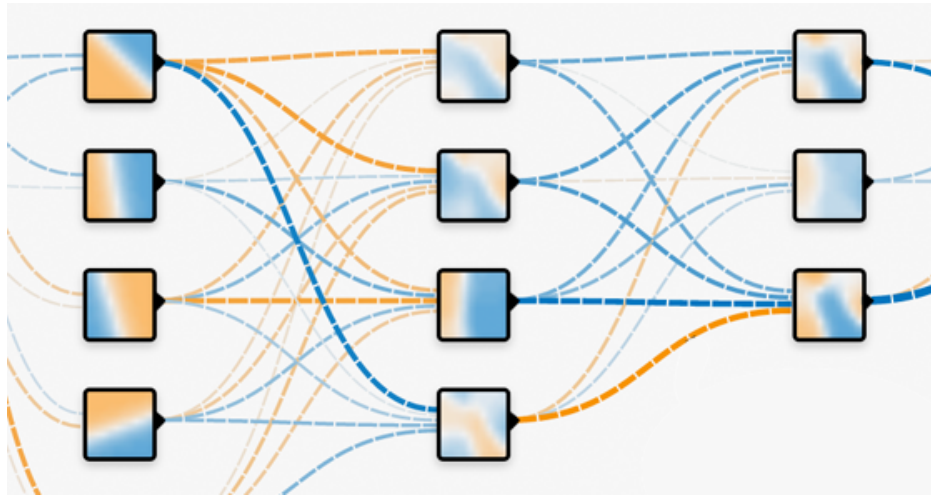


Figure 4.6: Neural network construction on Tensorflow: open-source platform for deep learning

tracking. But it is also utile as the input of a neural network training dataset. Full documentation has been provided, that users can be aware of any details that they are interested in. Problem is that there are some abnormal data in the database. It is probably caused by the mis-counting of original data as a consequence of blocking or overlapping. However, we cannot simply exclude the abnormal data since it is still relevant to its adjacent vehicles, the relative distance will be inaccurate then. It is the problem remains to be solved in the future.

Both the PTV Studio and SUMO simulator provides support on the OpenstreetMap Project [OpenStreetMap contributors \(2017\)](#). The OpenstreetMap (OSM) Project provides a detailed map file containing the road composition and public traffic route definition. Traffic infrastructure is much more sophisticated than one can imagine. It will a great pleasure for the researchers if there is a traffic roadmap which is ready to use. Rather than the map itself, the data generated by OSM is considered its primary output.

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

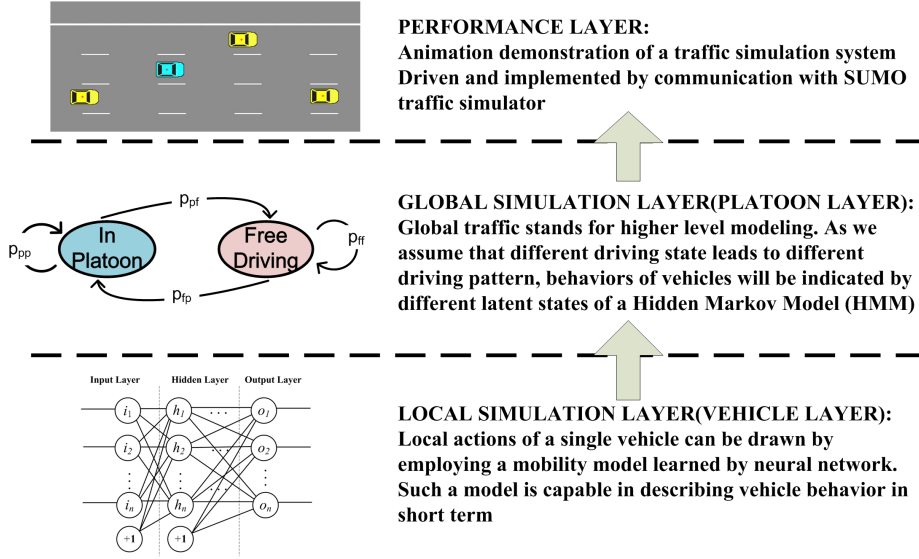


Figure 4.7: Our method in different layer

4.3 Scenario simulation

4.3.1 Highway scenario reproduction

To summarize our work in this thesis, we would like to present the traffic simulation of certain scenarios and the application of our proposed mobility model in these scenarios. Simulation in transportation is of big importance. It can study the models which are complicated for analytical treatment. Meanwhile, the simulation result provides an intuitive response of the input model. Under certain scenarios, impacts on different assignments of the model can be examined efficiently this way.

Differs from previous chapters, in which we focus on a single detailed method, simulation provided in this chapter will be related to all these methods, which means the simulation system is a multi-layer system (as shown in Figure 4.7). From bottom to top, in the local simulation layer, the mobility model control the behavior of a vehicle within the scenario. In the higher layer, driving state of the driver will be decided. The driving state will affect that which kind of mobility model will be used. Transitions of these states are described by an HMM. At the top, the animation layer is supported by the simulator platform. Based on the architecture of SUMO simulator and TraCI interface, we assign and update the real-time driving condition of vehicles.

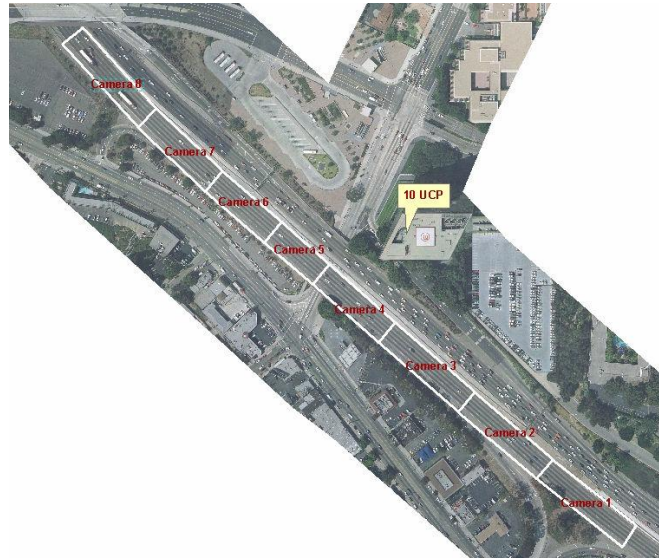


Figure 4.8: Road structure of the section that surveillance camera covers

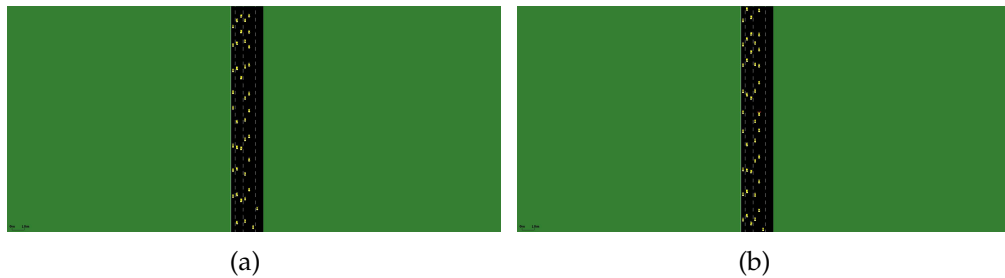


Figure 4.9: Highway reproduction scenario presented by SUMO

As for the first demonstration on detailed simulation, we would like to reproduce the scenario from our database. In NGSIM data collection, we employ a period of the record on US Highway 101, which describes a section of seven-lane highway in Los Angeles, California. This data was collected using video cameras mounted on a 36-story building, 10 Universal City Plaza, which is located adjacent to US Highway 101. Areas that are concerned under surveillance can be found as shown in Figure 4.8.

We reproduce the scenario, and the simulation result by SUMO can be found in Figure 4.9. The change in spacing and average velocity of the traffic flow are examined. As can be found in Figure 4.10, different curves represent the data generated by different models. The "-1" values in Figure 4.10(a) indicates that there are no leading vehicles so that it is not possible to measure the

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

spacing of this vehicle and we use a "-1" value to represent this situation. It can be concluded that all these models are capable of maintaining the spacing with preceding vehicles. Yet, a large difference appears in the average velocity. Due to the accumulation of errors, the velocity curves go further away from reality as time goes by. However, the curve of our NN-VMM remains a similar shape on changing of the curve. So when we examine the acceleration, we can see our curves follows the observed data quite well, meanwhile, the car-following cannot cover the reality.

The car-following model generates more "even" values as can be found in Figure 4.10(b), while despite the derivations on value, the velocity generated by our NN-VMM can have a similar tendency on changing of the velocity. Its derivatives can be used to measure this tendency. This can be found more specifically as in Figure 4.10(c). By reproducing the same traffic scenario, the credibility of our proposed MM will be proved.

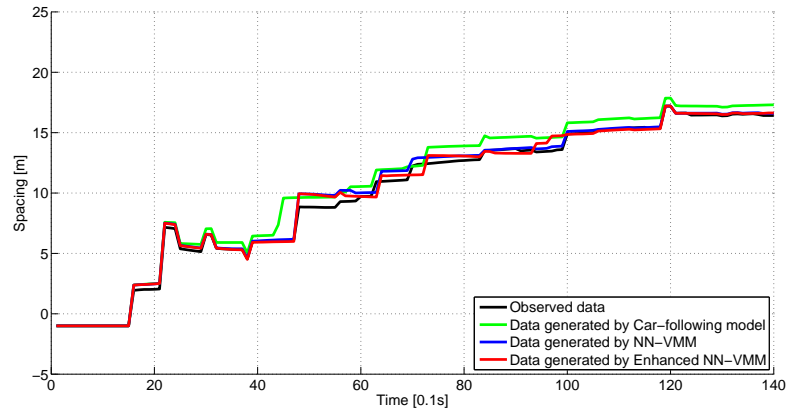
4.3.2 Scenario extension: MIXED NN-VMM/TP-AIM

Not limited to reproduce the scenario in the original database, the scenarios under simulated could be extended. With the help of OpenstreetMap project ([OpenStreetMap contributors \(2017\)](#)), a more complicated and realistic traffic scenario could be created. Under such a circumstance, our NN-VMM should be able to mimic the reality in some degrees. However, it could lose its accuracy because of the chosen dataset and the natural difference in driving property of different road condition.

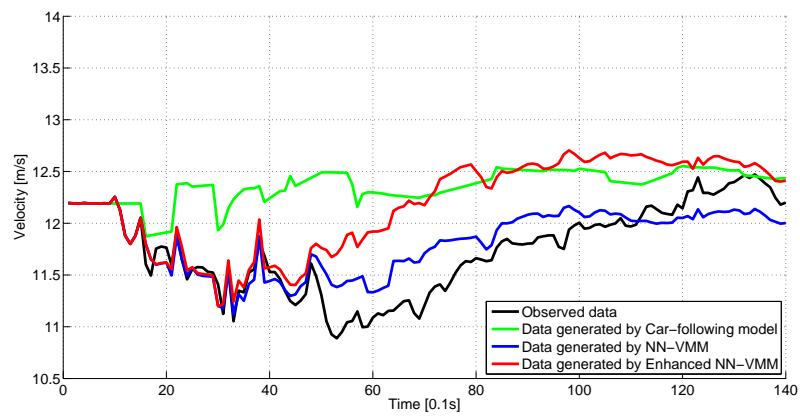
As an example, our NN-VMM is integrated into a previously described scenario as in [Liu *et al.* \(2018\)](#). It is an intersection traverse scenario using a trajectory planning based autonomous intersection management (TP-AIM). This blind traverse in TP-AIM is based on a calculation of the time occupation situation on the conflict zone (CZOT) inside an intersection: a CZOT arrangement algorithm. After the integration, a brief overview of the MIXED system can be seen as in Figure 4.11. Some modifications and specifications on the detailed protocol of this scenario simulation can be found as in Table 4.1.

The participant vehicles in the original experiment are divided into three groups in order to get across the intersection without collision. There are

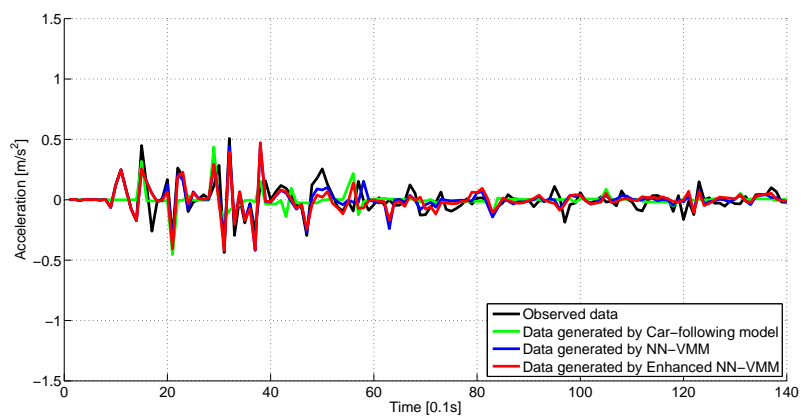
4.3 Scenario simulation



(a) Spacing distribution through time



(b) Average velocity distribution through time



(c) Acceleration distribution through time

Figure 4.10: Performance comparison from different models: original observed data, NN-VMM, Enhanced NN-VMM and Car-following Model

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

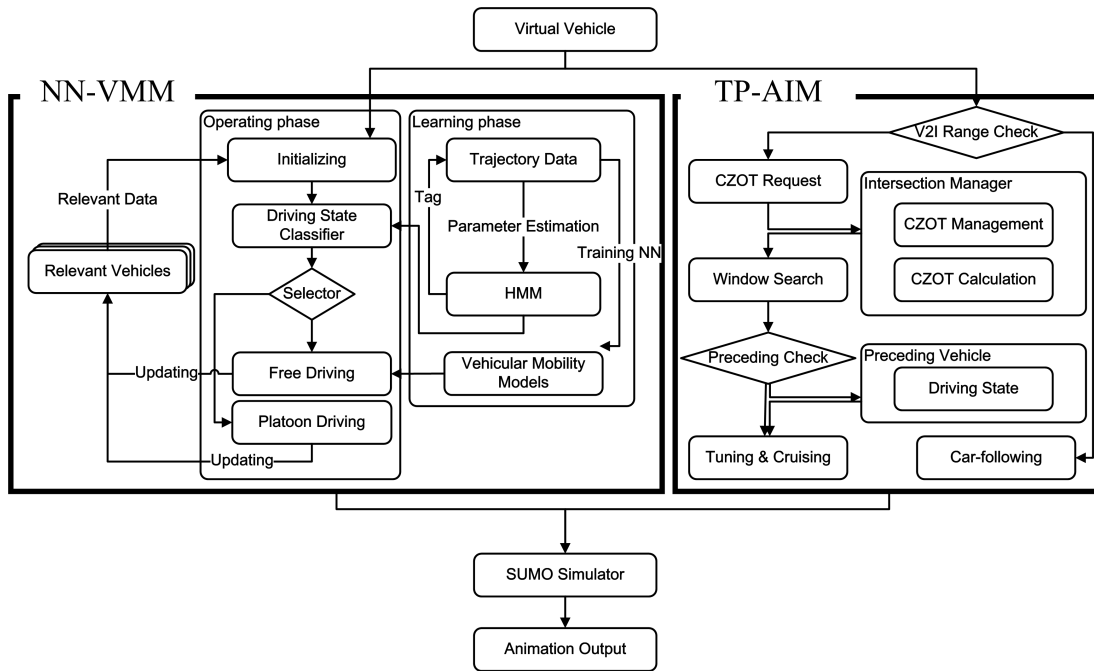


Figure 4.11: the MIXED simulation system: NN-VMM/TP-AIM

Table 4.1: Basic protocol of the simulation

Items	Details
Intersection construction	4-leg intersection as shown in Figure 4.12 with 3-lane for every direction
Leg-length	500 meters
V2I Communication range	150 meters
Participant vehicles	Randomly generated vehicle flow
Total number	632 vehicles
Go straight	208 vehicles
Turn left	213 vehicles
Turn right	211 vehicles
Simulation Time	98 seconds
step-length	0.2 second(s)
depart time of final vehicle	99 seconds
Vehicle flow composition	Randomly chosen from go straight vehicles
NN-VMM	Other vehicles
TP-AIM	
Free set	as defined in Chapter 4.3.2
Cruise set	as defined in Chapter 4.3.2
Tunning set	as defined in Chapter 4.3.2

- Cruising set: Vehicles should maintain their current velocities and the conflict zone occupation time (CZOT) information is calculated and stored at the intersection manager;
- Tuning set: Vehicles which currently have a conflict with some other vehicles are assigned with this set. A vehicle in this set must require the occupation time of all conflict zones from the intersection manager via communication, and it also requires position and velocity information from its nearest preceding vehicle. Then it can tune itself according to the information acquired to avoid rear-end collision and cross collisions;
- Free set: Vehicles that do not belong to the above two sets are assigned with this set.

The core of his experiment is the calculation of CZOT, whose nature is to solve the space-time conflict so that the vehicles can share the limited space resource inside the intersection. Detailed explanations can be found as in [Liu et al. \(2018\)](#). When we unfold the CZOT in time, the occupancy of a conflict zone within a duration can be seen as obstacles. Finding an entering window can be interpreted as finding a trajectory to bypass these obstacles. So a search window is launched to search across time and try to find space for the incoming vehicle. The shape of zones of the inserted vehicle is decided by vehicle's velocity and time when enters the intersection. During the searching process, all the found tuple of the entering state (v, t) having zero overlapped occupancy time are recorded as backup solutions, and the optimal solution would be assigned to trajectory planning at first.

Basic simulation by TP-AIM is based on the assumption of VANET implementation. V2I and V2V communications have been used to satisfy real-time changing of information, and the calculation associated information is gathered by this kind of communications. To be noticed that the assumption of implementation of VANET is a huge gap. It remains an ideal model in laboratory in the recent time. In order to loosen the assumption, we make our integration. In the NN-VMM/TP-AIM, VANET assumption is only applied to the none NN-VMM vehicles. We use this to simulate a mixed traffic flow: with a partial number of vehicles which is equipped with a V2X device and other vehicles that have not yet benefited this improvement.

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

In term of a vehicle, its static properties are assigned before the simulation (routes, starting lane, depart time, etc.). When it enters the scenario, it will be either assigned with NN-VMM or TP-AIM. This assignment is randomly, yet only vehicles go straight can be assigned with NN-VMM, since there is no turning trajectories in the original NGSIM database, and as a consequence NN-VMM cannot perform turning. Vehicles with NN-VMM cannot communicate with the other vehicles. They will travel and fulfill a traverse according to the surrounding situation detected by itself (local decision will be made based on local information). The other group of vehicles can communicate with each other just like the configuration of the original experiment. We assume that the intersection manager can achieve the occupation condition of the conflict zones by vehicles from both groups. Meanwhile, we've widened the searching window of CZOT since realistic vehicles are less predictable. Multiple vehicles within V2I range will demand CZOT in the same time, however, CZOT information will be assigned to them according to a priority sequence. For instance, in Figure 4.12, vehicles are approaching the intersection. Among them, different priorities are assigned. V_1 gets high score because it is a school bus and it is close to the intersection, V_2 gets higher score than V_3 and V_5 because it will turn left. Their demands of CZOT will be accepted according to their priorities. It is the same order of receiving intersection information and updating the CZOT at traffic manager.

Simulation results can be found as in Figure 4.14, where vehicles in blue are the vehicles controlled by our mobility model, vehicles in green, yellow and red are vehicles from cruising, free and turning set of the original method. Also as shown in Figure 4.13. Where multiple experiments are run with different composition of vehicle flow. Since our mobility model does not consider the CZOT situation, it does bring impact on the whole performance: the throughput remains at the level around 60% with and without the impact of our NN-VMM. Comparing with the traffic light scenario, whose throughput is 34%, the performance of TP-AIM does not decrease, which validate the blind crossing method in another way. However, another thing should be mentioned that as the increment of realistic vehicles within the flow, the CZOT management algorithm sometimes may not be able to find a valid solution, and collision happen this way.

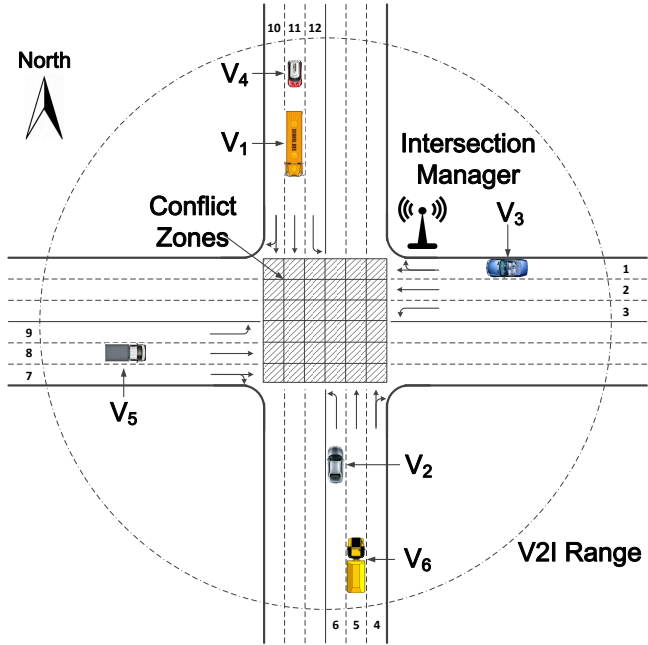


Figure 4.12: Road infrastructure of the under simulated intersection

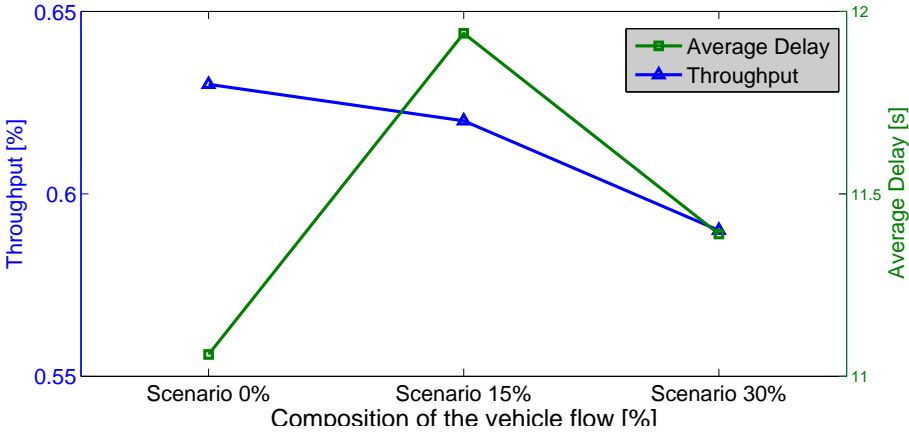


Figure 4.13: Impacts on throughput of different composition of the traffic flow

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

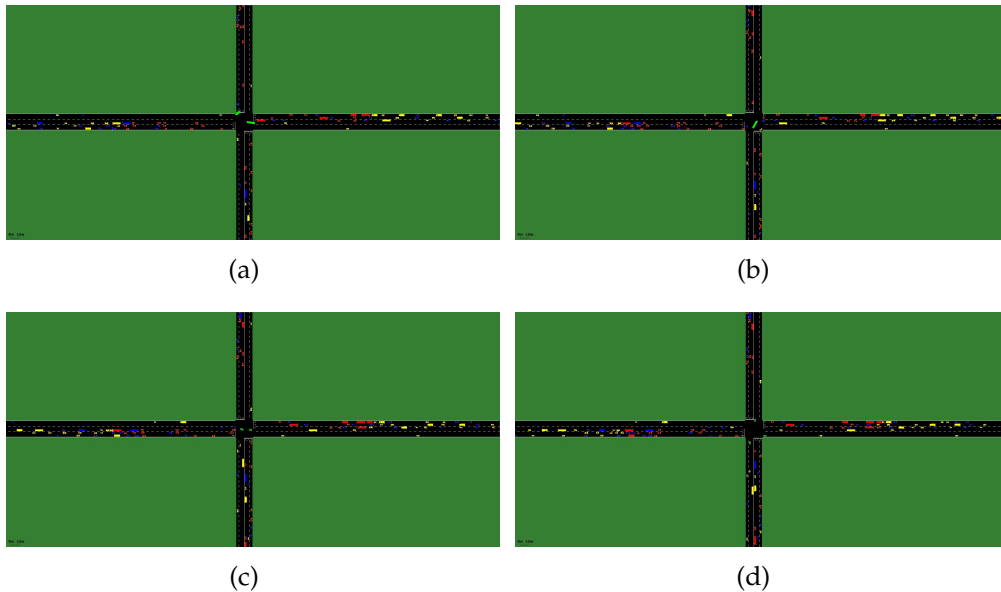


Figure 4.14: SUMO presentation of experiments on NN-VMM/TP-AIM

4.3.3 Applications in Transportation technology

Researches in transportation technology demand high realistic level vehicle models, which is just what our NN-VMM offers. This demand-supply relationship indicates that our NN-VMM may have serious applications in the field of transportation technology. In this section, we will demonstrate another scenario simulation to give a simple example of what can we do with this highly realistic level mobility model.

The simple example in this section will focus on the congestion issues in daily transportation. Plenty of researches have been done on solving or predicting traffic congestion, however, it still happens in our daily traffic. In this simulation, we choose the scenario that is very commonly seen in daily life, it's a highway on-ramp scenario. It can also be seen as an accident scenario or a road construction scenario that either sudden event happens to a vehicle or physically constraints force the vehicle to change its lane to another lane. The core of this simulation is that the congestion and queuing happens when vehicles are trying to import themselves into high-speed traffic.

As in this simulation, we built the scenario as a junction of two highway periods. Each period of the highway has a length of 150 meters. 4-lane highway is cut into 3 lanes here so that vehicles are forced to import themselves

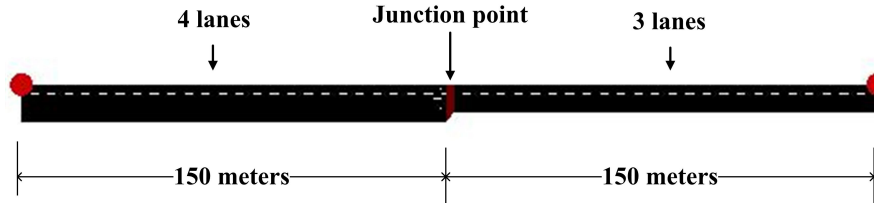


Figure 4.15: Highway structure of this simulation scenario

Table 4.2: Density of different flows

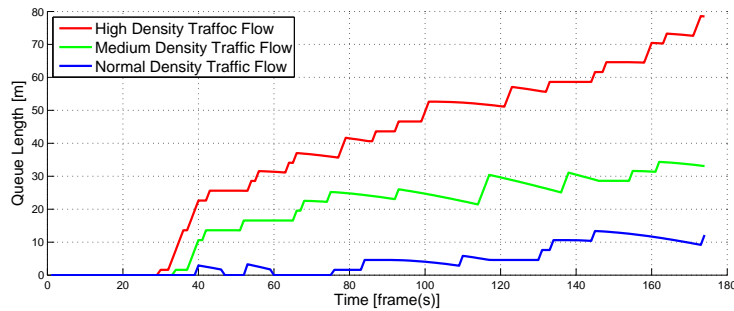
Flow Index	Participant vehicles	Density k (veh./300m)
Normal Density Flow	36	25
Medium Density Flow	58	40
High Density Flow	83	70

into other lanes and lane change behaviors will happen. The input of this simulation will be the different traffic flows, and we will examine the outputs: the generated queue length and other related flow properties. The participant vehicles are controlled by NN-VMM, yet in different flow densities. Simulation results can be seen as in Figure 4.16. Traffic flow density (k) is widely used in traffic flow theory to measure a traffic flow. It is defined as the number of vehicles per unit length of the roadway. The different traffic flows are initialized as follows:

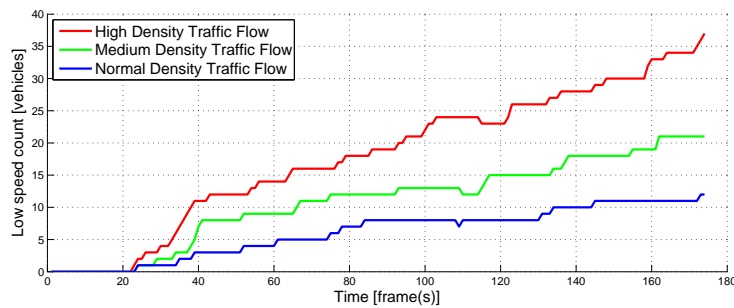
The simulation lasts for 300 frames, and the initialization of the scenario finishes in the 126th frames. So that we start to count the queue length from the 126th frame. From Figure 4.16(a), we can conclude that

- Queuing starts to appear in "normal density flow". The flow can digest the congestion in some degrees, but it cannot totally ease the congestion.
- Queuing lasts almost the whole lifetime of the simulation in "medium density flow". Differs from the "normal density flow", the queue has not been eased in this situation. However, it is able to maintain the length of the queue that it is rather stable and will not increase rapidly.

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION



(a) Queue length's change over time



(b) Low velocity vehicles' accumulation over time

Figure 4.16: Queue's forming over time

- In "high density flow", the queue length increases as the time goes by. There are constantly new members that join the queue yet few queue members get out of the queue.

A similar conclusion can also be made according to Figure 4.16(b). Instead of examining the queue length, we focus on the count number of low velocity vehicles (the vehicle that moves under the speed of $20km/h$) in this figure. Because sometimes, there is a vehicle that does not completely stop in a queue, but it travels at a very low speed and this may also cause congestion. The animation presentation of the simulation can be found as in Figure 4.17, where will use blue color to label the vehicles from the queue lane.

4.4 Discussion and Conclusion

In this chapter, we introduce the experiment platform for our NN-VMM and several scenario simulations that are built based on it. The platform combines

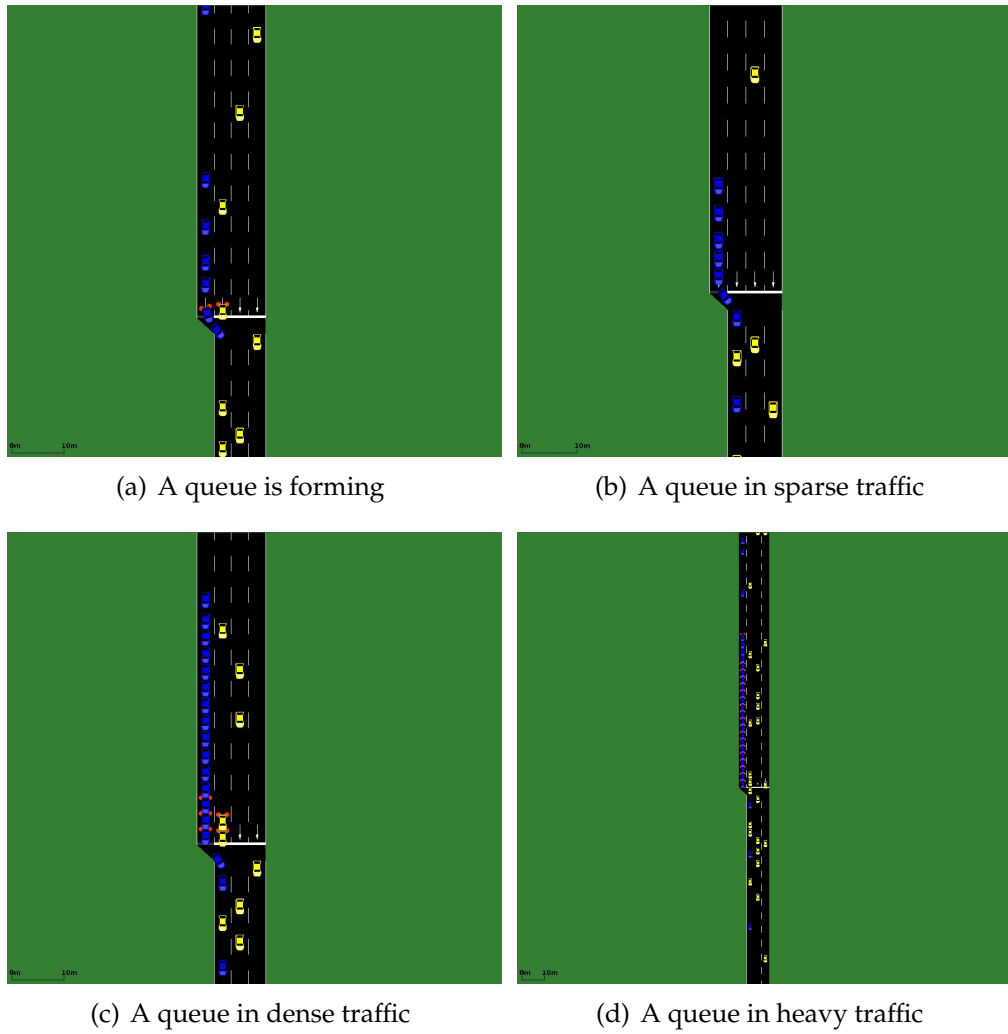


Figure 4.17: Traffic simulation for queue length estimation by SUMO

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

different parts of the whole methods. It associates with different conceptual and functional components. In order to clarify that, we use diagrams from UML to demonstrate our design of the platform and the related traffic simulation system. Since the timing and dependencies of different events and components are easily confused, we use the sequence diagram and state chart diagram to describe the dynamic property of the system.

The system has two major parts: offline learning part and online operating part. In the learning part, it can be further divided into the learning of HMM and learning of neural network, where the learning of neural network depends on the result of HMM's estimation and decoding. In the operating part, we should focus on the organization of vehicles: the calculation is accelerated with an efficient management on vehicles. Vehicles with our model operate in the virtual scenario. Their behaviors are calculated and passed through the interface to the presentation part. As the scale of simulation grows, it takes more time to compute. As a consequence, it loses its real-time property.

In this chapter, we also introduce the relevant open-source software packages. Different packages are in charge of different functional part of the whole system. Following researches can be built under the same framework as we employed in this thesis, which is a structure of Tensorflow-Traci4Matlab-SUMO.

We present three detailed simulations under this platform, a highway scenario, an intersection traverse scenario, and a queuing scenario. The former is the reproduction of the original observation scenario. Comparisons are constructed on this scenario to demonstrate the credibility of the proposed mobility model. It should be noticed that although the performance of our NN-VMM and enhanced NN-VMM is better than the car-following model, there is a gap between these models and the real-world data. What is gratifying is that our models follow the reality in tendency, and it is still much room for improvement in the future. The middle is an extension of previous work. Our NN-VMM is integrated with the TP-AIM to generate a mixed method NN-VMM/TP-AIM. The mixed method uses mixed flow to examine the performance of the proposed controlling method in a realistic condition. It is one typical application of mobility model, meanwhile, the mobility model can make a contribution in many other areas, such as its application in the latter. Our mobility model is used to observe a queuing phenomenon in a ramp structure

4.4 Discussion and Conclusion

and we estimate the queue length by validating the scenario with multiple traffic flows in different densities. Consequently, it appears that when the flow density is above 50 vehicles/300 meters, the congestion grows badly at the junction point, the queue length starts to increase. Similar simulations can be constructed for various situations to meet certain conditions. In this chapter, we only aim to claim its capability in applications of transportation technology of our work.

4. EXPERIMENT PLATFORM AND SCENARIO SIMULATION

Conclusions and Perspectives

This chapter is aiming at concluding the thesis. The results of each chapter is summarized here, then some limits and drawbacks in this work are analyzed, and several ideas to improve and extend this research are also proposed.

Conclusions

This thesis is dedicated to studying the vehicular mobility models with the help of deep learning methods, and the created model could make contributions in further control and optimization applications in Vehicular Ad Hoc Network, then in the intelligent transportation systems. Further researches on transportation can be built upon a platform with our proposed model.

In this thesis, the realistic level of the proposed mobility models has been discussed, which quite differs from the traditional models. Hypotheses have been made based on the current cognition and knowledge level on public transportation. It could be a great pleasure if researches in this thesis might be helpful and inspiring for other future researchers on related domains in intelligent transportation systems.

Detailed work has been presented as follows:

Chapter 1 mainly presents the background on intelligent transportation systems and motivation of this thesis. Problems exist in various aspects, such as the safety issues, efficiency issues and environmental friendly issues. Therefore, the concept of ITS comes into being consequently as a solution to all these challenges. Research on these issues are the common major concerns of ITS. State-of-the-art of the ITS is given also. To be specifically, the most attention goes to the related research domain in Vehicular Ad Hoc Network structure and the mobility models. According to the literature, mobility models have

CONCLUSIONS AND PERSPECTIVES

essential applications in not only the ITS, VANET, but also in behavior prediction for auto-navigation, in which the realism level plays a more important role than any other factors. Thus a conclusion has been made that our research locates on the research of vehicular mobility model which is especially against the limitation on previous models on realism issues.

Chapter 2 presents a detail probabilistic mobility model based on neural network learning using real-world collected trajectory traffic data collections. This model is the fundamental model of this thesis and will be improved as the discussion in this thesis goes deeper. The example-based model differs from a vehicular kinetic rule-based model, comes from data so that the data collection and its processing procedure is also introduced in this chapter. NN has been well studied over recent years, yet different NN structure may have its unique properties in handling corresponding problems. Since vehicular mobility models focus on the behavior of a vehicle in measurable physical attributes, which is not too specific as the throttle and brake controls or the pistons kinetic model within the engine, nor too rough as no existence of vehicle entities, we made adjustments of the collected data before training and the training result is demonstrated at the end of this chapter.

Chapter 3 has enhanced the mobility model proposed in Chapter 2 with an HMM. In order to address this enhancement, an adaptation method of traditional mobility models is in need. Thus, the traditional mobility models have been introduced first. Comparing with the state-of-the-art in Chapter 1, complementary introductions are given on categorizations and major restrictions on movements. The mobility models have been developed from simple movements of a single entity to temporal-spatial dependency included movements of a group of vehicles. However, these models cannot be directly used in a probabilistic model such as HMM, so that an adaptation is necessary. Examples from 1 dimension adaptation to practical 2 dimensions car-following models adaptation are reviewed. The obtained model after adaptation will preserve the major statistical feature of the model, meanwhile expressed as in a probabilistically form. Then, the demand of specification on data can be addressed by the three basic problems in HMM. We estimated the parameters with the partial knowledge on the model: the adapted traditional model. This

knowledge ensures the HMM to be initialized near its global maximum. Certain practical problems are also reviewed, such as the multiple independent observation sequences and the scaling problem when calculating the model in value in a computer. The estimation result is given in this chapter. Along with the estimated matrices in HMM, a procedure of labeling back to the database can be done by solving the decoding problem. With the labeled specified data, an enhanced mobility model can be obtained by the same method proposed in Chapter 2.

Chapter 4 has explained the whole design of a traffic simulation system that combines all the elements described in former chapters. This is a platform that we employed to implement experiments on. The design is described by UML diagrams both in static modeling and in dynamic modeling. Moreover, certain implementation issues have also been discussed, including commercial usage and open-source software packages. We would like to use some open source software to support and connect every different component of the whole simulation, which includes SUMO simulator, Tensorflow learning testbed, and TraCI interface for the communicating through different programs. Finally simulations are given, accompany with three detailed scenarios to demonstrate the performance of our mobility model in simulation and its applications in a higher level.

Perspectives

Based on the results given by this thesis, several perspectives should be considered:

- The data collection can be extended. As it has been emphasized multiple times in this thesis, the data collection plays an important role in this data-driven model. The employed highway vehicle trajectory data determines that the learned mobility model shows better performance in mimicking a highway scenario. It is possible to extend the model to urban traffic situation if corresponding data collection can be found. Problem is that little attention has been paid to the surveillance trajectory data in urban traffic. On the other hand, several attributes in NGSIM database are not fully considered in implementation, such as the vehicle type, lane

CONCLUSIONS AND PERSPECTIVES

identification, etc. Integration of these attributes can enrich the scope of the proposed model even with the same current used data collection.

- It could be interesting to combine our model with a more powerful machine learning method. The development of machine learning is rapid. Back to the FFNN, the testing performance is satisfactory by 10^{-5} in MSE. However, it goes down to 10^{-3} for an RNN normally, since it is much more difficult to train this more complicated network structure before the deep learning methods are proposed. Therefore, it is rational for us to looking forward to a more effective learning method in the future that our learning based performance can be improved as a consequence.
- The representation of HMM can be alternative. In this thesis, the HMM is built with discrete observations. In theory, the observation can also be continuous. Thus, the emission of HMM, which is described by the pmf in discrete case, is described by the probability density function (pdf) of the continuous variables.
- The mathematical vehicle model could be enriched. An example of the adaptation algorithm in Chapter 3 is given with the Gipps car-following model. This certain car-following model itself is a representative of the family of car-following. Other kinds of car-following models, or to be more general, other kinds of vehicular mobility models are also possible in doing an adaptation for HMM. However, since different variables may be involved, that the inference procedure will be different. Moreover, when decomposing to find the meta-event of the probability, different sampling methods may lead to different results. In the future, a mathematical proof of the adaptation algorithm can be studied, so that more general models can be guaranteed usable in HMM after an adaptation.
- The resource consumption of communication can be further reduced. According to the employed simulator (SUMO) in our experimental platform, a veh-graph structure among vehicles and corresponding updating algorithms are proposed in this thesis to reduce the communication cost between the simulator and controller when conducting certain scenario simulations. To be noticed, at the peak, there are more than a hundred

CONCLUSIONS AND PERSPECTIVES

vehicles in the scene. They all need to communicate and update their location information through the same simulator and controller, which bring great computational resource consumption. However, for the vehicle behavior prediction in ego-centric applications, only the information of several surrounding vehicles needs to be calculated and updated, which will greatly shorten the entire running time.

CONCLUSIONS AND PERSPECTIVES

Résumé Étendu en Français

Introduction

Les systèmes de transport intelligents ont acquis un grand intérêt pour la recherche ces dernières années. Les exigences en matière de transport intelligent se situent sous différents aspects, tels que la sécurité du trafic, l'efficacité du trafic, etc. Parmi celles-ci, un trafic réaliste joue un rôle important, mais il n'a pas reçu suffisamment d'attention. Cette thèse est consacrée à l'étude de la simulation du trafic au niveau microscopique, et propose des modèles de mobilité des véhicules, qui ont pour but de récupérer les flux de trafic virtuel haut niveau réaliste. Basé sur des données de trajectoire de trafic collectées, les modèles de mobilité proposés extraient le modèle de conduite et fourniront un flux de circulation convaincant. Cela peut avoir des utilisations essentielles dans des domaines tels que le réseau ad hoc de véhicules (VANet), la conduite autonome, etc.

Motivation de la recherche

À la suite de la modernisation et de l'urbanisation, les systèmes de transport deviennent beaucoup plus grands et plus lourds, ce qui apporte des défis pour la maintenance et l'organisation d'un tel système de transport. Le nombre de véhicules par habitant (voir Figure 1) a maintenant atteint un nombre insensé. Selon les recherches statistiques, le nombre de véhicules automobiles par millier d'habitants atteint 910 aux États-Unis en 2017 et plus de 500 dans la plupart des pays européens. Ce nombre de Chine est presque cinq fois plus important qu'il ne l'était il y a dix ans (2006).

RÉSUMÉ ÉTENDU EN FRANÇAIS

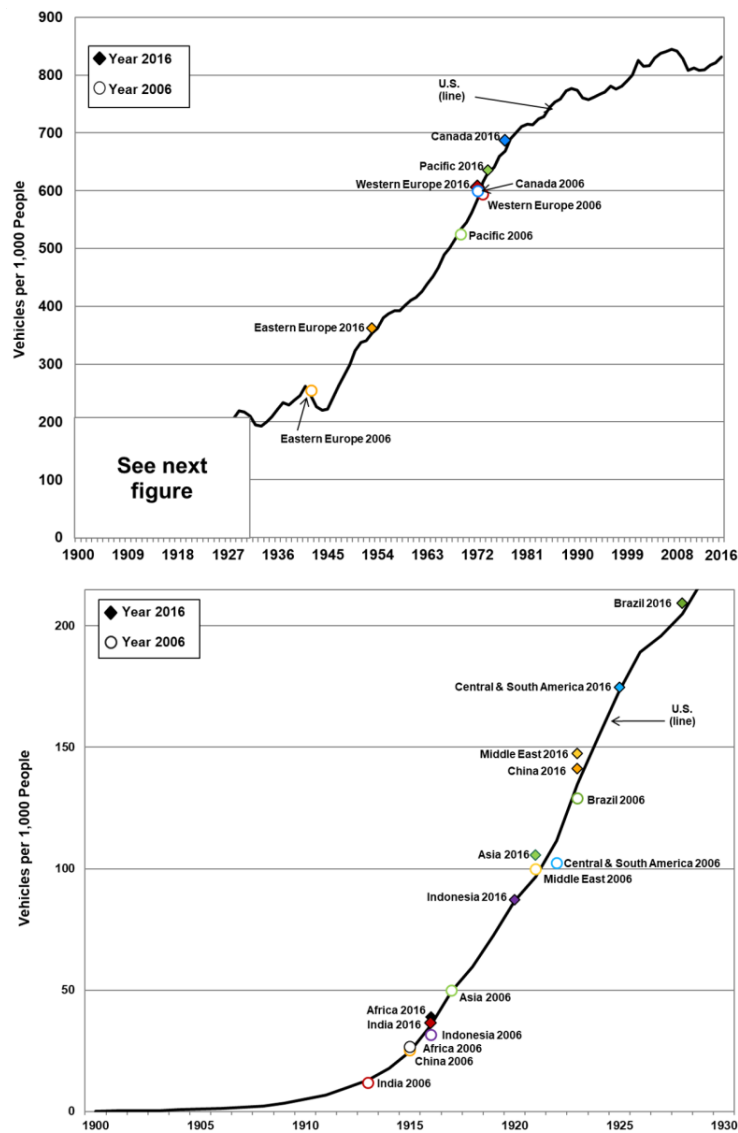


Figure 1: Véhicules par millier d'habitants: États-Unis et autres pays (en 2006 et 2016)

(Source <http://www.greencarcongress.com/2018/10/20181010-fotw.html>)

Dans ce contexte, le concept "systèmes de transport intelligents (STI)" est proposée. Les STI peuvent être largement définis comme l'application des technologies de pointe, telles que les communications, les capteurs, et l'informatique, aux système de transport, afin de fournir des informations en temps réel pour améliorer la sécurité, l'efficacité et le confort, en même temps de réduire l'impact sur l'environnement. En outre, en raison de l'augmentation du système de transport, la prévision réaliste du trafic virtuel est également devenue un sujet clé des STI. Plus précisément, il s'agit de l'acquisition d'un modèle de mobilité suffisamment réaliste et réaliste. Le modèle de mobilité de VANet décrit le comportement des véhicules. Il est proposé d'étudier l'impact de la mobilité des nœuds sur la communication au sein du réseau.

Habituellement, les modèles de mobilité traditionnels sont conçus pour traiter des situations dans certaines conditions prédéfinies. Il ne convient pas aux situations inconnues, qui sont très courantes dans le trafic quotidien à l'heure actuelle, en raison de l'extension du transport. Par conséquent, cette thèse utilise une méthode pilotée par les données pour obtenir le modèle de mobilité de véhicule réaliste correspondant. L'acquisition du modèle de mobilité final contient deux aspects

Modèle de mobilité basé sur un réseau de neurones

Dans le premier aspect, nous utilisons un réseau de neurones pour apprendre simplement le modèle de conduite à partir des données de trajectoire observées.

Si vous établissez une simulation de scénario, puis examinez et comparez la vitesse moyenne avec les données réelles collectées, il est évident que la méthode basée sur des exemples présente un avantage en termes de niveau réaliste par rapport à la méthode basée sur des règles. En effet, c'est parce que la méthode basée sur des règles, par exemple un modèle suivi de voiture, concerne les contraintes prédéfinies, tandis que la méthode basée sur des exemples, par exemple, un modèle appris par le réseau de neurones, se concentre sur les démonstrations, qui reflètent le comportement dans le monde réel. L'apprentissage du réseau de neurones a été discuté pendant des décennies. Le réseau de neurones est largement utilisé dans la classification et la prédiction.

L'algorithme de rétropropagation du gradient garantit un calcul efficace lors du calcul des dérivées d'erreur de sortie sur les poids. Cependant, le dégradé disparaît lorsque le réseau est constitué de couches profondes. De plus, en raison de la limitation de la capacité de calcul, il est particulièrement difficile d'accomplir l'entraînement d'un réseau de neurones lorsque le réseau comporte un grand nombre de démonstrations d'entrée. Des avancées ont été réalisées grâce à l'application de la méthode d'apprentissage profond, qui permet de l'entraînement d'un réseau de neurones à plusieurs couches et avec une quantité énorme de démonstrations d'entrées. Les entrées sont divisées en mini-lots. Au lieu de calculer toutes les entrées, un lot est choisi au hasard pour chaque itération. L'algorithme du gradient stochastique avec mini-lots est beaucoup plus rapide que l'algorithme du gradient typique dans le cas d'un grand nombre d'entrées. En outre, la "batch normalisation" est utilisée pour stimuler le gradient entre les couches. La normalisation entre les couches masquées permet de ne pas perdre le dégradé lors de la traversée de plusieurs couches.

La collecte de données utilisée provient de la base de données "Simulation de prochaine génération (SIMPG)". Elle contient les attributs d'un véhicule au niveau microscopique. Les données de trajectoire peuvent être récupérées après la reconstitution du scénario. Un graphique de véhicule est construit pour réduire les coûts de communication entre le modèle et le simulateur SMOU (Simulation de la mobilité urbaine). La conception du système de simulation de trafic (voir Figure 2) basée sur ce modèle de mobilité appris indique l'architecture du système et la relation entre les différents composants du système. Les résultats de la simulation expérimentale montrent que l'idée d'un modèle de mobilité piloté par les données basé sur un réseau de neurones est réalisable, mais que des améliorations supplémentaires peuvent être apportées pour améliorer les performances.

Amélioration du modèle de mobilité par intégration avec MMC

Le modèle de mobilité obtenu par apprentissage sur réseau neuronal présente des inconvénients pour la simulation à long terme en raison de l'accumulation

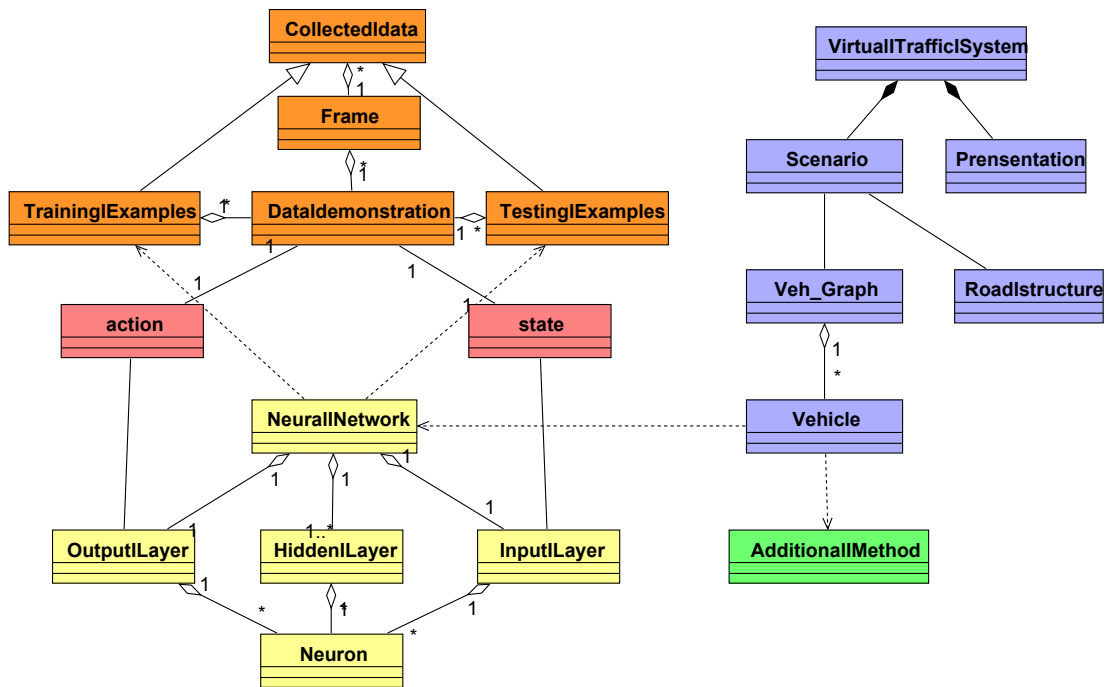


Figure 2: Conception de système dans un diagramme de classes UML

d'erreurs. Une amélioration possible peut être apportées en utilisant des données plus précises pour l'entraînement au réseau de neurones. On pense qu'il existe plusieurs styles de conduite dans le scénario à partir de la collecte de données utilisée. Différents styles devraient conduire à différents modèles de mobilité. Cependant, les données de différents styles de conduite sont mélangées pour l'entraînement d'un réseau de neurones.

Afin de distinguer les états de conduite, un modèle de Markov caché (MMC) a été introduit. Dans un MMC :

- L'état de conduite est caché et non observable.
- Les états de conduite transitent dans le temps de manière probabiliste.
- Les états de conduite déterminent les attributs observables de manière probabiliste.

Des recherches sur le MMC ont été effectuées en résolvant les trois problèmes suivants:

- problème d'évaluation : Connaissant l'automate, calculer la probabilité d'une séquence particulière.
- problème de décodage : Connaissant l'automate, trouver la séquence la plus probable d'état (caché) ayant conduit à la génération d'une séquence de sortie donnée.
- problème d'estimation : Étant donné une séquence de sortie, retrouver l'ensemble d'états le plus probable et les probabilités des sorties sur chaque état.

Dans notre cas, une estimation du MMC sera faite en premier. En utilisant l'algorithme Baum-Welch avec la collecte de données, la matrice de transition et la matrice d'émission de MMC peuvent être estimées. Ensuite, les états de conduite correspondants peuvent être identifiés en résolvant un problème de décodage à l'aide de l'algorithme de Viterbi. Enfin, l'entraînement au réseau de neurones se fera avec les données étiquetées, ce qui améliorera les performances puisque les données proviennent d'un état de conduite spécifié.

L'organigramme de la simulation avec modèle de mobilité améliorée (voir Figure 3) est divisé en deux phases : la phase d'entraînement et la phase d'exploitation. L'estimation MMC et le décodage de la collecte de données sont effectués dans la phase d'entraînement avant l'entraînement du réseau de neurones. En phase d'exploitation, l'état de conduite change de manière probabiliste à chaque pas de temps. Des modèles de mobilité correspondants de différents états de conduite seront utilisés. Une révision à partir d'un modèle cinétique du simulateur SMOU affinera la trajectoire de sortie et, enfin, la simulation sera effectuée par le simulateur.

Algorithme d'adaptation et estimation MMC avec connaissances partielles

Habituellement, la définition d'état MMC provient d'une procédure PHD-MMC (Processus Hiérarchique de Dirichlet-Modèle de Markov Caché), dans laquelle le centre de mise en cluster ayant le score le plus élevé des données sera traité comme état MMC. Cependant, la définition de l'état de cette manière semble

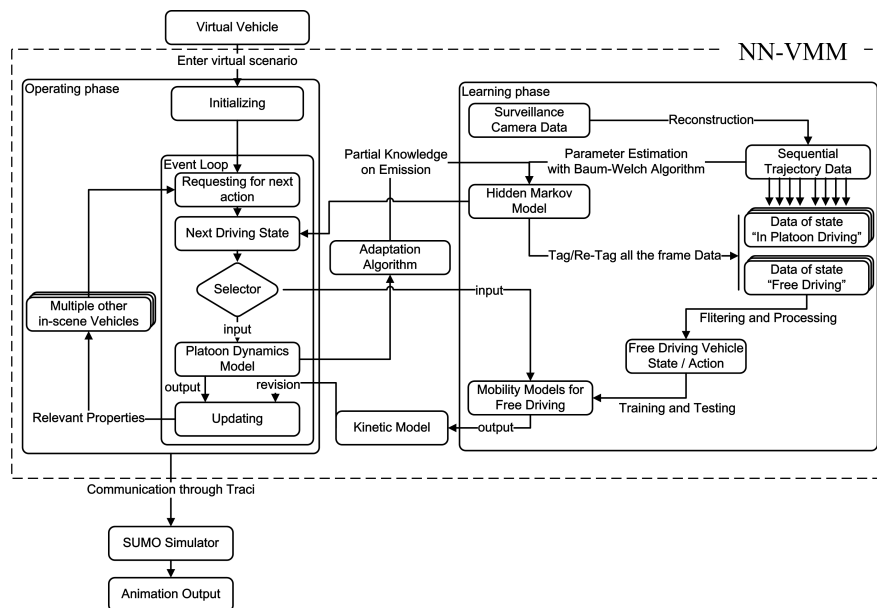


Figure 3: L'organigramme de la simulation avec modèle de mobilité améliorée

être assez abstraite : les états ne sont pas liés à un style de conduite réel dans le monde réel. Au contraire, certains états de conduite dans notre MMC sont désignés.

Selon la base de données SIMPG, nous définissons un état de "conduite en peloton" dans lequel les véhicules se comportent selon le modèle de suivi de voiture de Gipps. Les autres appartiennent à l'état "conduite libre" et les données de l'état "conduite libre" seront envoyées au réseau de neurones pour l'apprentissage du modèle de mobilité.

L'avantage de cette définition est que la connaissance partielle du MMC estimé est acquise puisque l'un des états est désigné. A noter que l'initialisation de MMC est très importante et que la connaissance de l'estimation de MMC peut aider à initialiser MMC afin d'éviter un minimum local. Cependant, le modèle de suivi de voiture n'est pas présenté de manière à pouvoir être introduit dans un MMC, de sorte qu'une adaptation du modèle est nécessaire. L'algorithme d'adaptation (voir Figure 4) comprend une décomposition du méta-événement et un échantillonnage des variables d'entrée. Nous avons examiné les attributs du modèle de suivi de voiture de Gipps. La vitesse et l'espacement ne sont pas conditionnellement indépendants. En conséquence, une variable conjointe est utilisée pour représenter l'attribut observable. Après

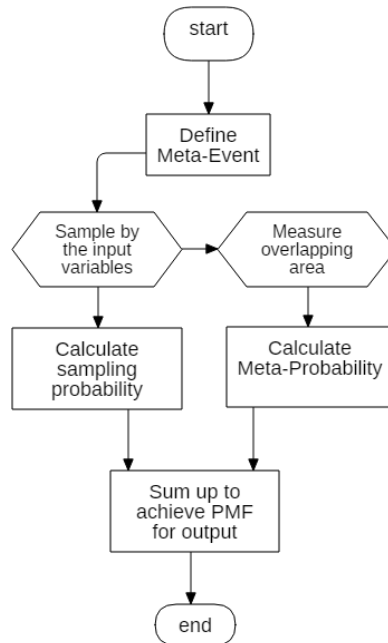


Figure 4: Procédure de l’algorithme d’adaptation

l’adaptation du modèle, la représentation de probabilité du modèle pour l’émission de MMC est obtenue. L’estimation peut alors être réalisée avec cette connaissance partielle.

Conclusions et Perspectives

Cette thèse est consacrée à l’étude des modèles de mobilité des véhicules à l’aide de méthodes d’apprentissage profond. Plusieurs simulations de scénarios ont été construites pour examiner les performances des modèles proposés dans différentes applications.

Contributions

- Premièrement, un modèle de mobilité piloté par les données basé sur le réseau de neurones est proposé. Le modèle peut produire un flux véhiculaire virtuel réaliste et prometteur dans un scénario autoroutier. En fonction de l’environnement de simulation, un graphique de véhicule et son

algorithmes d'organisation sont proposés pour réduire les coûts de communication entre le simulateur et notre modèle de mobilité.

- Deuxièmement, des améliorations ont été apportées à l'aspect des données d'entrée. Un MMC est introduit pour représenter la procédure de conduite avec différents styles de conduite. Dans notre cas, le problème peut être présenté en résolvant les problèmes fondamentaux de MMC : le problème d'estimation et le problème de décodage.
- Troisièmement, un algorithme d'adaptation est proposé pour obtenir la connaissance partielle d'un état de conduite désigné. Les modèles de suivi de voiture traditionnels peuvent être adaptés pour répondre à l'exigence de forme de représentation de probabilité de MMC. Avec le modèle adapté, le MMC peut être initialisé plus précisément et les ajustements sur l'équation d'estimation ont également été effectués.
- Enfin, nous présentons la plate-forme expérimentale pour la simulation de trafic. Plusieurs simulations de scénarios sont construites pour démontrer la capacité du modèle de mobilité proposé. La comparaison entre différents modèles et les données du monde réel dans le scénario de reconstruction d'une autoroute est utilisée pour évaluer le niveau réaliste de notre modèle. Notre modèle proposé peut également être utilisé pour fournir un flux de trafic réaliste afin de valider d'autres modèles et de réaliser certaines applications de transport, telles que l'estimation de la longueur de la file d'attente.

Perspectives

- La collecte de données peut être étendue. L'extension des données utilisées dans le trafic urbain peut faire ressortir une meilleure performance du modèle de mobilité dans des scénarios de trafic urbain. D'autre part, plusieurs attributs de la base de données SIMPG n'ont pas été pleinement pris en compte. Ce faisant, la performance dans les comportements de changement de voie sera améliorée.
- Il pourrait être intéressant de combiner notre modèle avec une méthode d'apprentissage automatique plus puissante. Le développement du

RÉSUMÉ ÉTENDU EN FRANÇAIS

réseau de neurones et l'apprentissage en profondeur sont rapides. Par conséquent, il est rationnel pour nous de rechercher une méthode d'apprentissage plus efficace à l'avenir que notre performance basée sur l'apprentissage puisse être améliorée en conséquence.

- Le modèle de véhicule mathématique peut être enrichi. Un exemple est donné par le modèle de suivi de voiture Gipps. En outre, plusieurs autres modèles de suivi de voiture ou autres modèles de mobilité peuvent être utilisés pour l'adaptation à l'émission de MMC. Il convient de noter que l'adaptation repose sur la boucle fermée des variables impliquées. À l'avenir, une preuve mathématique de l'algorithme d'adaptation peut être étudiée, de sorte que des modèles plus généraux puissent être garantis utilisables dans MMC après une adaptation.

Bibliography

- ABADI, M., BARHAM, P., CHEN, J., CHEN, Z., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., IRVING, G., ISARD, M. *et al.* (2016). Tensorflow: A system for large-scale machine learning. In *OSDI*, vol. 16, 265–283. [95](#), [106](#)
- ARTIMY, M.M., ROBERTSON, W. & PHILLIPS, W.J. (2004). Connectivity in inter-vehicle ad hoc networks. In *Electrical and Computer Engineering, 2004. Canadian Conference on*, vol. 1, 293–298, IEEE. [15](#)
- ARTIMY, M.M., PHILLIPS, W.J. & ROBERTSON, W. (2005). Connectivity with static transmission range in vehicular ad hoc networks. In *Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual*, 237–242, IEEE. [15](#)
- BAI, F. & HELMY, A. (2004). A survey of mobility models. *Wireless Adhoc Networks. University of Southern California, USA*, **206**, 147. [16](#), [55](#)
- BASKAR, L.D., DE SCHUTTER, B., HELLENDORN, J. & PAPP, Z. (2011). Traffic control and intelligent vehicle highway systems: a survey. *IET Intelligent Transport Systems*, **5**, 38–52. [10](#), [16](#)
- BAUM, L.E. (1968). Growth functions for transformations on manifolds. *Pac. J. Math.*, **27**, 211–227. [82](#)
- BAUM, L.E. & EAGON, J.A. (1967). An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, **73**, 360–363. [82](#)
- BAUM, L.E. & PETRIE, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, **37**, 1554–1563. [85](#)

BIBLIOGRAPHY

- BLUM, J.J., ESKANDARIAN, A. & HOFFMAN, L.J. (2004). Challenges of inter-vehicle ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, **5**, 347–351. 15
- BORNE, P., POPESCU, D., FILIP, F.G. & STEFANOIU, D. (2013). *Optimization in Engineering Sciences: Exact Methods*. John Wiley & Sons. 83
- BOTTOU, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186, Springer. 38
- BRACKSTONE, M. & MCDONALD, M. (1999). Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, **2**, 181–196. 29
- BROCH, J., MALTZ, D.A., JOHNSON, D.B., HU, Y.C. & JETCHEVA, J. (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 85–97, ACM. 57
- BROGGI, A., MEDICI, P., ZANI, P., COATI, A. & PANCIROLI, M. (2012). Autonomous vehicles control in the vislab intercontinental autonomous challenge. *Annual Reviews in Control*, **36**, 161–171. 17
- BUREAU, U.C. (2018). Maps data tiger products. <https://www.census.gov/geo/maps-data/data/tiger.html>. 20
- CAMP, T., BOLENG, J. & DAVIES, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, **2**, 483–502. 56
- CHANDLER, M. (2016). Google, baidu, tesla gunning self-driving car development. www.investors.com/news/technology. 18
- CHARALAMBOUS, P. & CHRYSANTHOU, Y.L. (2016). Data-driven crowd evaluation. *Simulating Heterogeneous Crowds with Interactive Behaviors*, 179. 21
- CHEN, X., LI, L. & ZHANG, Y. (2010). A markov model for headway/spacing distribution of road traffic. *IEEE Transactions on Intelligent Transportation Systems*, **11**, 773–785. 80

- DAVIES, A. (2014). Audi's self-driving car hits 150 MPH on an F1 track. www.wired.com/2014/10/audis-self-driving-car-hits-150-mph-f1-track/. 18
- DAVIES, A. (2015). Baidu's self-driving car has hit the road. www.wired.com. 17
- DAVIES, V.A. *et al.* (2000). *Evaluating mobility models within an ad hoc network*. Master's thesis, Citeseer. 57
- DE BRUIN, D., KROON, J., VAN KLAVEREN, R. & NELISSE, M. (2004). Design and test of a cooperative adaptive cruise control system. In *Intelligent Vehicles Symposium, 2004 IEEE*, 392–396, IEEE. 19
- DE SCHUTTER, B. & DE MOOR, B. (1998). Optimal traffic light control for a single intersection. *European Journal of Control*, 4, 260–276. 20
- FANCHER, P. (1998). Intelligent cruise control field operational test. final report. volume i: Technical report. 60
- FIGUEIREDO, L., JESUS, I., MACHADO, J.T., FERREIRA, J. & DE CARVALHO, J.M. (2001). Towards the development of intelligent transportation systems. In *Intelligent Transportation Systems*, vol. 88, 1206–1211. 10
- FORNEY, G.D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61, 268–278. 84
- GERAERTS, R. & SCHAGER, E. (2010). Stealth-based path planning using corridor maps. In *Computer Animation and Social Agents*. 14
- GIPPS, P.G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15, 105–111. 29, 57, 63
- GRAVES, A., MOHAMED, A.R. & HINTON, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*, 6645–6649, IEEE. 41
- GRZYBEK, A., DANOY, G. & BOUVRY, P. (2012). Generation of realistic traces for vehicular mobility simulations. In *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, 131–138, ACM. 11

BIBLIOGRAPHY

- GUIZZO, E. (2011). How google's self-driving car works. *IEEE Spectrum Online*, October, **18**. [17](#)
- HAGAN, M.T., DEMUTH, H.B., BEALE, M.H. & DE JESÚS, O. (1996). *Neural network design*, vol. 20. Pws Pub. Boston. [27](#)
- HÄRRI, J., FILALI, F., BONNET, C. & FIORE, M. (2006). Vanetmobisim: generating realistic mobility patterns for vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, 96–97, ACM. [20](#)
- HARRI, J., FILALI, F. & BONNET, C. (2009). Mobility models for vehicular ad hoc networks: a survey and taxonomy. *IEEE Communications Surveys & Tutorials*, **11**. [55](#)
- HARTENSTEIN, H. & LABERTEAUX, K. (2010). *VANET: vehicular applications and inter-networking technologies*, vol. 1. Wiley Online Library. [14](#), [19](#)
- HOCHREITER, S. & SCHMIDHUBER, J. (1997). Long short-term memory. *Neural computation*, **9**, 1735–1780. [41](#)
- HONG, X., GERLA, M., PEI, G. & CHIANG, C.C. (1999). A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 53–60, ACM. [59](#)
- HOUENOU, A., BONNIFAIT, P., CHERFAOUI, V. & YAO, W. (2013). Vehicle trajectory prediction based on motion model and maneuver recognition. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 4363–4369, IEEE. [22](#)
- INTEL (2015). Building an intelligent transportation system with the the internet of things (iot). <http://www.intel.cn/content/www/cn/zh/internet-of-things>. [11](#)
- IOFFE, S. & SZEGEDY, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. [39](#)

- JARDOSH, A., BELDING-ROYER, E.M., ALMERTH, K.C. & SURI, S. (2003). Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, 217–229, ACM. [20](#)
- JIANG, B. & FEI, Y. (2015). Traffic and vehicle speed prediction with neural network and hidden markov model in vehicular networks. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, 1082–1087, IEEE. [27](#)
- KARNADI, F.K., MO, Z.H. & LAN, K.C. (2007). Rapid generation of realistic mobility models for vanet. In *Wireless communications and networking conference, 2007. WCNC 2007. IEEE*, 2506–2511, IEEE. [20](#)
- KENDZIORRA, A., WAGNER, P. & TOLEDO, T. (2016). A stochastic car following model. *Transportation Research Procedia*, **15**, 198–207. [60](#)
- KOROSEC, K. (2016). Tesla: This is our most significant step towards safe self-driving cars. <http://fortune.com/2016/02/09/tesla-self-parking/>. [18](#)
- KRAJZEWICZ, D., ERDMANN, J., BEHRISCH, M. & BIEKER, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, **5**, 128–138. [45](#), [95](#), [106](#)
- KRAUSS, S. (1998). *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. Ph.D. thesis, Universitat zu Koln. [29](#)
- KRAUSS, S., WAGNER, P. & GAWRON, C. (1997). Metastable states in a microscopic model of traffic flow. *Physical Review E*, **55**, 5597. [29](#)
- LAMY, D. & BORNE, P. (1993). Neural networks initialization. In *Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, vol. 4, 491–495, IEEE. [34](#)
- LERNER, A., CHRYSANTHOU, Y. & LISCHINSKI, D. (2007). Crowds by example. In *Computer Graphics Forum*, vol. 26, 655–664, Wiley Online Library. [28](#)

BIBLIOGRAPHY

- LEVINSON, J., ASKELAND, J., BECKER, J., DOLSON, J., HELD, D., KAMMEL, S., KOLTER, J.Z., LANGER, D., PINK, O., PRATT, V. *et al.* (2011). Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 163–168, IEEE. [16](#)
- LI, F. & WANG, Y. (2007). Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular technology magazine*, **2**, 12–22. [10](#)
- LIANG, B. & HAAS, Z.J. (1999). Predictive distance-based mobility management for pcs networks. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 1377–1384, IEEE. [58](#)
- LIU, B. (2016). *Control and optimization for intelligent transportation systems in vicinity of intersections*. Ph.D. thesis, Ecole Centrale de Lille. [102](#)
- LIU, B. & EL KAMEL, A. (2016). V2x-based decentralized cooperative adaptive cruise control in the vicinity of intersections. *IEEE Transactions on Intelligent Transportation Systems*, **17**, 644–658. [10](#), [20](#)
- LIU, B., SHI, Q., SONG, Z. & EL KAMEL, A. (2018). Trajectory planning for autonomous intersection management of connected vehicles. *Simulation Modelling Practice and Theory*, Accepted. [110](#), [113](#)
- LIU, H.X., WU, X., MA, W. & HU, H. (2009). Real-time queue length estimation for congested signalized intersections. *Transportation research part C: emerging technologies*, **17**, 412–427. [10](#)
- LU, X.Y., HEDRICK, J.K. & DREW, M. (2002). Acc/cacc-control design, stability and robust performance. In *American Control Conference, 2002. Proceedings of the 2002*, vol. 6, 4327–4332, IEEE. [19](#)
- LUO, M., EL KAMEL, A. & GONG, G. (2011). Uml-based design of intelligent vehicles virtual reality platform. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 115–120, IEEE. [102](#)
- LV, Y., DUAN, Y., KANG, W., LI, Z., WANG, F.Y. *et al.* (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intelligent Transportation Systems*, **16**, 865–873. [21](#), [22](#)

- MA, X., TAO, Z., WANG, Y., YU, H. & WANG, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, **54**, 187–197. [21](#), [27](#)
- MACKAY, D.J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press. [27](#)
- MAHMUD, K. & TOWN, G.E. (2016). A review of computer tools for modeling electric vehicle energy requirements and their impact on power distribution networks. *Applied Energy*, **172**, 337–359. [45](#), [105](#)
- MONTANINO, M. & PUNZO, V. (2015). Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transportation Research Part B: Methodological*, **80**, 82–106. [21](#)
- MONTEMERLO, M., BECKER, J., BHAT, S., DAHLKAMP, H., DOLGOV, D., ETTINGER, S., HAEHNEL, D., HILDEN, T., HOFFMANN, G., HUHNKE, B. *et al.* (2008). Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, **25**, 569–597. [17](#)
- NGSIM (2017). Us highway 101 dataset. <https://www.fhwa.dot.gov/publications/research/operations/07030/>. [47](#), [106](#)
- NGUYEN, D. & WIDROW, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, 21–26, IEEE. [34](#)
- OPENSTREETMAP CONTRIBUTORS (2017). Planet dump retrieved from <https://planet.osm.org> . <https://www.openstreetmap.org>. [107](#), [110](#)
- OSORIO, R., PEÑA, M., LÓPEZ-JUÁREZ, I., SAVAGE, J. & LEFRANC, G. (2013). Using background and segmentation algorithms applied in mobile robots. *IFAC Proceedings Volumes*, **46**, 135–140. [20](#)
- PARK, Y. & MIN, D. (2015). Distributed traffic simulation using dds-communication based HLA for V2X. In *2015 Seventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 450–455, IEEE. [14](#)

BIBLIOGRAPHY

- PEÑA-CABRERA, M., LOPEZ-JUAREZ, I., RIOS-CABRERA, R. & CORONA-CASTUERA, J. (2005). Machine vision approach for robotic assembly. *Assembly Automation*, **25**, 204–216. [16](#)
- PERNG, J.W., WEN, Y.H., CHANG, W.J. & CIOU, M.Y. (2011). Design and implementation of an intelligent vehicle driving controller. In *2011 8th Asian Control Conference*, 1111–1116, Kaohsiung. [10](#)
- PUNZO, V., BORZACCHIELLO, M.T. & CIUFFO, B. (2011). On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (ngsim) program data. *Transportation Research Part C: Emerging Technologies*, **19**, 1243–1262. [21](#)
- RABINER, L.R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257–286. [92](#)
- RAJAMANI, R. (2011). *Vehicle dynamics and control*. Springer Science & Business Media. [19](#)
- REYNOLDS, C.W. (1999). Steering behaviors for autonomous characters. [26](#), [28](#)
- ROYER, E.M., MELLIAR-SMITH, P.M. & MOSER, L.E. (2001). An analysis of the optimum node density for ad hoc mobile networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 3, 857–861, IEEE. [57](#)
- RUDER, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*. [38](#)
- RUMELHART, D.E., HINTON, G.E. & WILLIAMS, R.J. (1986). Learning representations by back-propagating errors. *nature*, **323**, 533. [33](#)
- RUSSELL, S. (2006). DARPA grand challenge winner: Stanley the robot! *Popular Science*. [16](#)
- SAHA, A.K. & JOHNSON, D.B. (2004). Modeling mobility for vehicular ad-hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 91–92, ACM. [20](#)
- SHEN, D. (2008). Some mathematics for hmm. *Massachusetts Institute of Technology*. [92](#)

- SHKVARKO, Y., ATOCHE, A.C. & TORRES-ROMAN, D. (2011). Near real time enhancement of geospatial imagery via systolic implementation of neural network-adapted convex regularization techniques. *Pattern Recognition Letters*, **32**, 2197–2205. [37](#)
- SONG, X., KANASUGI, H. & SHIBASAKI, R. (2016). Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *IJCAI*, vol. 16, 2618–2624. [21](#), [22](#)
- SUKTHANKAR, R., HANCOCK, J. & THORPE, C. (1998). Tactical-level simulation for intelligent transportation. *Mathematical and computer modelling*, **27**, 229–242. [16](#)
- THRUN, S., MONTEMERLO, M., DAHLKAMP, H., STAVENS, D., ARON, A., DIEBEL, J., FONG, P., GALE, J., HALPENNY, M., HOFFMANN, G. *et al.* (2006). Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, **23**, 661–692. [16](#)
- TIAPRASERT, K., ZHANG, Y., WANG, X.B. & ZENG, X. (2015). Queue length estimation using connected vehicle technology for adaptive signal control. *IEEE Transactions on Intelligent Transportation Systems*, **16**, 2129–2140. [10](#)
- TRACI (2018). Traffic control interface. <http://www.sumo.dlr.de/wiki/TraCI>. [106](#)
- URMSON, C., ANHALT, J., BAGNELL, D., BAKER, C., BITTNER, R., CLARK, M., DOLAN, J., DUGGINS, D., GALATALI, T., GEYER, C. *et al.* (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, **25**, 425–466. [17](#)
- VAHIDI, A. & ESKANDARIAN, A. (2003). Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE transactions on intelligent transportation systems*, **4**, 143–153. [18](#)
- VITERBI, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, **13**, 260–269. [84](#), [95](#)

BIBLIOGRAPHY

- WANG, S. (2004a). On the intermittence of routing paths in vehicle-formed mobile ad hoc networks on highways. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, 803–809, IEEE. 14
- WANG, S.Y. (2004b). Predicting the lifetime of repairable unicast routing paths in vehicle-formed mobile ad hoc networks on highways. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 4, 2815–2819, IEEE. 14
- WHO (2018). Road traffic injuries. <http://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>. 10
- WILLIAMS, R.J. & ZIPSER, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, **1**, 270–280. 41
- WOLINSKI, D. (2016). *Microscopic crowd simulation: evaluation and development of algorithms*. Ph.D. thesis, Université Rennes 1. 21
- XIA, C. & EL KAMEL, A. (2016). Neural inverse reinforcement learning in autonomous navigation. *Robotics and Autonomous Systems*, **84**, 1–14. 28
- XIAO, L. & GAO, F. (2010). A comprehensive review of the development of adaptive cruise control systems. *Vehicle System Dynamics*, **48**, 1167–1192. 19
- XU, D., HE, X., ZHAO, H., CUI, J., ZHA, H., GUILLEMARD, F., GERONIMI, S. & AIOUN, F. (2017). Ego-centric traffic behavior understanding through multi-level vehicle trajectory analysis. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 211–218, IEEE. 22, 80
- XU, Q. & SENGUPTA, R. (2003). Simulation, analysis, and comparison of acc and cacc in highway merging control. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 237–242, IEEE. 19
- YOUSEFI, S., MOUSAVI, M.S. & FATHY, M. (2006). Vehicular ad hoc networks (vanets): challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, 761–766, IEEE. 14

- YU, E. & CHEN, C.R. (1993). Traffic prediction using neural networks. In *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM'93., IEEE*, 991–995, IEEE. [27](#)
- YU, Y., EL KAMEL, A. & GONG, G. (2014a). Modeling and simulation of overtaking behavior involving environment. *Advances in Engineering Software*, **67**, 10–21. [26](#)
- YU, Y., EL KAMEL, A., GONG, G. & LI, F. (2014b). Multi-agent based modeling and simulation of microscopic traffic in virtual reality system. *Simulation Modelling Practice and Theory*, **45**, 62–79. [45](#)
- ZHANG, J. & EL KAMEL, A. (2018). Traffic simulation with virtual traffic flow. In *International Conference on Advanced Intelligent Systems and Informatics*, Accepted, Springer. [30](#)
- ZHANG, J. & EL KAMEL, A. (2018). Virtual traffic simulation with neural network learned mobility model. *Advances in Engineering Software*, **115**, 103–111. [88](#), [95](#), [97](#)
- ZHANG, J., EL KAMEL, A. & LEFRANC, G. (2018). Enhanced mobility model using hidden markov model for traffic simulation. *Advances on Engineering Software*, Under review. [77](#)
- ZHANG, L. & LENDERS, P. (2002). Morlet wavelet based human head and face boundary extraction. In *Proceedings of Fifth Asian Conference on Computer Vision, Melbourne*, 801–806, Citeseer. [20](#)
- ZHAO, D., DAI, Y. & ZHANG, Z. (2012). Computational intelligence in urban traffic signal control: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**, 485–494. [10](#)
- ZIEGLER, C. (2015). Volvo will run a public test of self-driving cars with 100 real people in 2017. www.theverge.com. [18](#)
- ZONOOZI, M.M. & DASSANAYAKE, P. (1997). User mobility modeling and characterization of mobility patterns. *IEEE Journal on selected areas in communications*, **15**, 1239–1252. [57](#)

BIBLIOGRAPHY

Modèles de Mobilité de Véhicules par Apprentissage Profond dans les Systèmes de Transport Intelligents

Résumé: Les systèmes de transport intelligents ont acquis un grand intérêt pour la recherche ces dernières années. Alors que la simulation réaliste du trafic joue un rôle important, elle n'a pas reçu suffisamment d'attention. Cette thèse est consacrée à l'étude de la simulation du trafic au niveau microscopique et propose des modèles de mobilité des véhicules correspondants. À l'aide de méthodes d'apprentissage profond, ces modèles de mobilité ont fait leurs preuves avec une crédibilité prometteuse pour représenter les véhicules dans le monde réel. D'abord, un modèle de mobilité basé sur un réseau de neurones piloté par les données est proposé. Ce modèle provient de données de trajectoires du monde réel et permet de mimer des comportements de véhicules locaux. En analysant les performances de ce modèle de mobilité basé sur un apprentissage de base, nous indiquons qu'une amélioration est possible et proposons ses spécifications. Un MMC est alors introduit. La préparation de cette intégration est nécessaire, ce qui comprend un examen des modèles de mobilité traditionnels basés sur la dynamique et l'adaptation des modèles "classiques" à notre situation. Enfin, le modèle amélioré est présenté et une simulation de scénarios sophistiqués est construite pour valider les résultats théoriques. La performance de notre modèle de mobilité est prometteuse et des problèmes de mise en œuvre sont également discutés.

Mots-clés: Apprentissage profond, Systèmes de transport intelligents, Modèle de mobilité des véhicules, Réseau de neurones, Modèle de Markov caché, Simulation du trafic.

Deep learning based vehicular mobility model for intelligent transportation systems

Abstract: The intelligent transportation systems gain great research interests in recent years. Although the realistic traffic simulation plays an important role, it has not received enough attention. This thesis is devoted to studying the traffic simulation in microscopic level, and proposes corresponding vehicular mobility models. Using deep learning methods, these mobility models have been proven with a promising credibility to represent the vehicles in real-world. Firstly, a data-driven neural network based mobility model is proposed. This model comes from real-world trajectory data and allows mimicking local vehicle behaviors. By analyzing the performance of this basic learning based mobility model, we indicate that an improvement is possible and we propose its specification. An HMM is then introduced. The preparation of this integration is necessary, which includes an examination of traditional dynamics based mobility models and the adaptation method of "classical" models to our situation. At last, the enhanced model is presented, and a sophisticated scenario simulation is built with it to validate the theoretical results. The performance of our mobility model is promising and implementation issues have also been discussed.

Keywords: Deep learning, Intelligent transportation systems, Vehicular mobility model, Neural network, Hidden Markov model, Traffic simulation.

BIBLIOGRAPHY
