UNIVERSITY OF TECHNOLOGY SYDNEY
School of Electrical and Data Engineering

Modeling, Analysis and Application of Big Traffic Data for Intelligent Transportation Systems

by

Peibo Duan

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2019

Certificate of Authorship/Originality

I, Peibo Duan declare that this thesis, is submitted in fulfillment of the requirements for the award of doctorate, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature: Signature removed prior to publication.

Date: 12th, January, 2020

© Copyright 2019 Peibo Duan

ABSTRACT

Modeling, Analysis and Application of Big Traffic Data for Intelligent Transportation Systems

by

Peibo Duan

Intelligent Transportation System (ITS), an integrated system of people, roads, and vehicles by utilizing information and communications technology, has emerged as an efficient way of improving the performance of transportation systems, enhancing travel security, and providing more choice to travelers. Recently, it has been seen that the big data era for ITS is coming due to the wide use of traffic detectors like traffic cameras and GPSs. These traffic detectors can collect various types of traffic data that significantly contribute to the development of ITS, which has the benefit of the public with convenient and safe travel.

With big traffic data, data-driven methods provide powerful and theoretical support for data modeling, analysis, and applications. However, existing methods still suffer from some shortcomings. First, traffic predictors usually use black-box methods to capture the spatiotemporal correlation between traffic. As a result, it reduces the flexibility of predictors due to the time-varying spatial-temporal correlation caused by frequent variation of road conditions. Second, it is impossible to cover all urban areas with traffic detectors. Thus, data absence and data sparsity have an essential impact on the reliability of travel state monitoring in a large road network. Lastly, most big data applications are based on the centralized method for processing and analyzing data, which consume more time and computational resources, optimal decision making. These make research on big traffic data in ITS become both exciting and essential.

In this thesis, a physically intuitive approach is developed for short-term traffic

flow prediction that captures the time-varying spatiotemporal correlation between traffic, mainly attributed to the road network topology, travel speed, and trip distribution. Experimental results demonstrate its superior accuracy and lower computational complexity compared with its counterparts. After that, a novel methodology is presented to estimate link travel time distributions (TTDs) using end-to-end (E2E) measurements detected by the limited traffic detectors. The experimental results show that the estimated results are in excellent agreement with the empirical distributions. Lastly, a distributed scheme is proposed for taxi cruising route recommendations based on taxi demands predicted by the proposed Graph Convolutional Network (GCN) based method. Experiment and simulation are both implemented. Experimental results validate the accuracy of the proposed taxi demand predictor. Simulation results indicate that our proposed taxi recommendation scheme is better than its counterparts in the aspects of minimizing the number of vacant taxis and maximizing the global revenue of taxi drivers.

Dissertation directed by Professor Guoqiang Mao
School of Electrical and Data Engineering

Acknowledgements

The completion of this dissertation has been possible with the inspiration and encouragement from many people, to whom I am greatly indebted.

First of all, I overwhelmingly pay my immeasurable appreciation and deepest gratitude to my supervisor, Professor Guoqiang Mao, for his persistent guidance, valuable recommendations, generous advice never-ending patience and ongoing support. I feel extremely fortunate to be mentored by him during my Ph.D. candidature.

I am also grateful to other members in the research team of Prof. Guoqiang, for their making a friendly working environment, and for their support, great assistance, and valuable advice to my research.

I sincerely acknowledge the deepest love and ongoing support from my beloved families during my studies. I am grateful to my mother, my grandparents, and my father in heaven, for their encouragement and blessing.

Finally, but mostly, I wish to express my deepest gratitude to my friends, Mencheng Gao, Haoqi Zhou, Yiyang Jiang, Ying Wang, Ying He, and Yang Yang, who were always standing by me in my hard times during this work, especially in the innumerable days and nights fulfilled with anxiety, self-denial and hopelessness. Thanks for all the joy and beautiful memories they brought to my life. Although for some of them, we may never see each other again, I consider my self extremely fortunate to have all of them in the journal of my life.

Peibo Duan

Sydney, Australia

July 2019

List of Publications

The following is a list of publications in refereed journals and conference proceedings produced during my Ph.D. candidature. In some cases, the journal papers contain material overlapping with the conference publications.

Journal Papers

J-1. **Peibo Duan**, Changsheng Zhang, Guoqiang Mao, and Bin Zhang, "Applying distributed constraint optimization approach to the user association problem in heterogeneous networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 16961707, 2018.

J-2. **Peibo Duan**, Guoqiang Mao, Weifa Liang, Degan Zhang, "A Unified Spatio-temporal Model for Short-term Traffic Flow Prediction," **Accepted** by *IEEE Transactions on Intelligent Transportation System*, NOv. 2018.

J-3. **Peibo Duan**, Guoqiang Mao, Baoqi Huang, Jun Kang, "Estimation of Link Travel Times Distribution with Limited Traffic Detectors," **Accepted** by *IEEE Transactions on Intelligent Transportation System*, 2019.

Conference Papers

C-1. **Peibo Duan**, Guoqiang Mao, Shangbo Wang, Changsheng Zhang and Bin Zhang, "STARIMA-based Traffic Prediction with Time-varying Lags," *IEEE 19th Int. Conf. on Intelligent Transportation Systems*, pp. 1610-1615, 2016.

C-2. **Peibo Duan**,Guoqiang Mao, Wenwei Yue, and Shangbo Wang, "A Trade-off between Accuracy and Complexity: Short-term Traffic Flow Prediction with Spatio-temporal Correlations," *IEEE 21th Int. Conf. on Intelligent Transportation Systems*, pp. 1658-1663, 2018.

C-3. **Peibo Duan**, Guoqiang Mao, Changsheng Zhang, and Jun Kang, "A Unified STARIMA based Model for Short-term Traffic Flow Prediction," *IEEE 21th Int. Conf. on Intelligent Transportation Systems*, pp. 1652-1657, 2018.

C-4. **Peibo Duan**, Guoqiang Mao, Baoqi Huang, and Jun Kang, "Estimating Link Travel Time Distribution Using Network Tomography Technique," **Accepted** by *IEEE 22th Int. Conf. on Intelligent Transportation Systems*,Jun. 2019.

Contents

List of Figures

Abbreviation

ARIMA - AutoRegressive Integrated Moving Average

ANN - Artificial Neural Network

ARIMAX - ARIMA with EXogenous variables

BPNN - Back Propagation Neural Network

BN - Bayesian Network

BTMS - Bluetooth Traffic Monitoring System

BFS - Breadth First Search

CNN - Convolutional Neural Network

CDF - Cumulative Density Function

DTMC - Discrete Time Markov Chains

DCRNN - Diffusion Convolutional Recurrent Neural Network

EM - Expectation Maximization

E2E - End-to-End

FCL-Net - Fusion Cnvolutional Long short-term memory Network

GNN - Graph Neural Network

GMM - Gaussian Mixture Model

GCN - Graph Convolutional Network

GN - GirvanNewman

HPP - Homogeneous Poisson Process

HW - Holt Winters

ITS - Intelligent Transportation System

KARIMA - Kohonen ARIMA

KNN - K Nearest Neighbor

KDE - Kernel Density Estimator

LSTM - Long Short Term Memory

MAC - Media Access Control

MDP - Markov Decision Process

MIP - Mixed Integer Programming

POI - Position of Interest

PDF - Probability Density Function

RNN - Recursive Neural Network

STARIMA - Space-Time ARIMA

SVR - Support Vector Regression

SARIMA - Seasonal ARIMA

STW-KNN - Spatio-Temporal Weighted KNN

SVD - Singular Value Decomposition

STL - Seasonal and Trend decomposition using Loess

STGCNN - Spatial-Temporal Graph Convolutional Neural Network

STM-GCN - Spatio-Temporal multi-GCN

SACF - Spatial AutoCorrelation Function

SPACF - Spatial Partial AutoCorrelation Function

SMS - Space Mean Speed

TTP - Traffic Transition Probability

TMS - Time Mean Speed

TTD - Travel Time Distribution

Chapter 1

Introduction

With the increasing demands of transportation development, traffic problems such as congestion have increased to such an extent that enhancing the present infrastructure of roads is no longer a cost-effective and efficient solution to these problems. Moreover, the enhancement to the present infrastructure may bring large expense in terms of finance and labor, and hence require much more time, which meanwhile results in more traffic congestions and bottlenecks. To cope with these challenges, Intelligent Transportation System (ITS) has attracted a lot of attention. It builds an integrated system of people, roads and vehicles by utilizing information and communications technology to improve road performance, reduce accidents, optimize fuel consumption, and enable multimodal transport. Due to the application of various traffic sensors, ITS has accumulated huge and complex traffic data that has put forward new requirements to the management and processing of information. As a result, data-driven methods provide a powerful and flexible data analysis and processing function to mine the traffic system's real-time and the comprehensive traffic model. It can be used in traffic management and control, and hence improves the service level of the ITS.

1.1 Research Background

Fig.1.1 shows the architecture of conducting big data analytics in ITS, which consists of three layers, respectively data collection layer, data analytics layer, and application layer. In the following sections, we briefly illustrate these three layers, respectively.

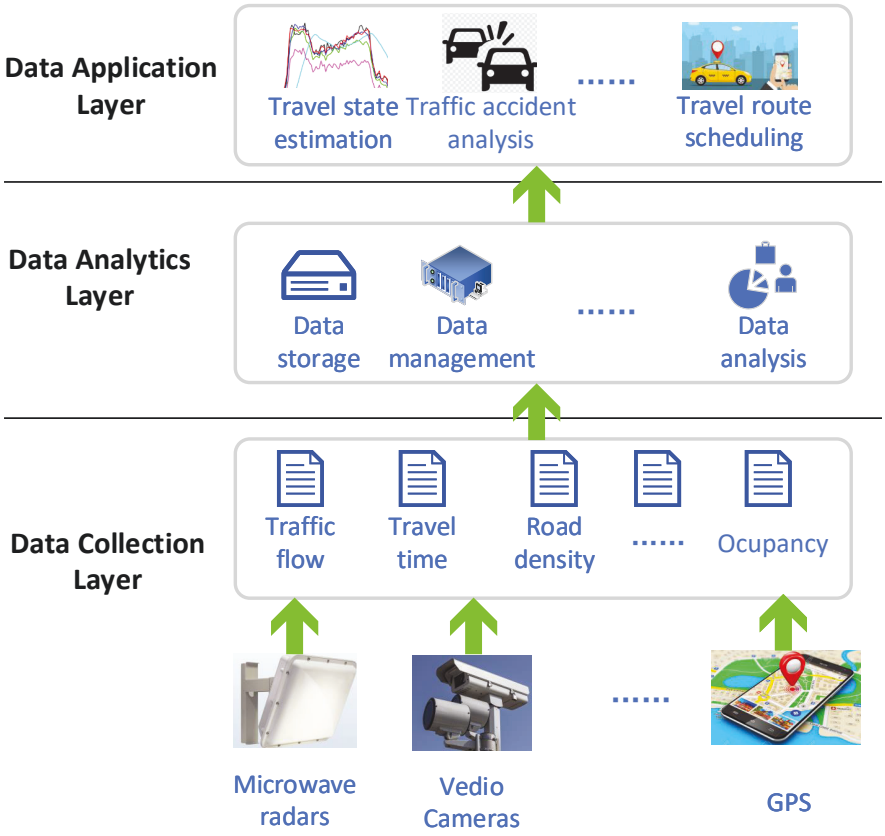


Figure 1.1 : The three layer architecture of conducting big data analytics in ITS

1.1.1 Methods of Traffic Data Collection

The methods of traffic data collection can be broadly split into two categories: the stationary and non-stationary detector based methods [48]. Stationary detector based methods indicate that the traffic data are measured with detectors configured along the roadside or in the roads. We describe the most important traffic detectors, hereafter (Please see Table 1.1).

Non-stationary detectors collect real-time traffic data by locating the vehicle via positioning systems, mainly including GPS and cellular-based systems.

With these traffic detectors, a large amount of traffic data is collected and processed into various forms for different stakeholders (like travelers and transportation administrators). This data mainly includes traffic flow, travel time, travel speed,

Table 1.1 : Introduction of widely used stationary traffic detectors

Traffic detector	Principle	Advantage	Limitation	Typical Research
Pneumatic road tubes	They are placed across the road lanes. The vehicles are detected when the air pressure in the tubes are changed when a vehicle tire passes over the tube. The data is then processed and recorded by a counter configured at the side of the road. They are usually used to count the number of vehicles.	It has low cost and is easy to maintain.	It has inaccurate axle counting when traffic volumes are high. Beside, they are sensitive to the temperature.	[57, 87]
Piezoelectric sensors	The sensors are located in a groove along the roadway surface of the lanes monitored. They measure the weight and speed of the vehicles by detecting the difference between the electrodes caused by the variation of surface charge density of the material when the vehicles pass over the sensors.	It has small size and can be made in any desired shape. It also has good frequency response.	It is sensitive to the temperature.	[52]
Magnetic loops	They are embedded in the roads in a square formation to generate a magnetic field. The data is then transmitted to a counter placed on the side of the road.	It is small and canbe mounted close to the ground.	It needs high voltage and current variable capacitors.	[16, 17]
Passive magnetic sensors	The sensors are deployed under or on top of the roadbed. The main function aims at counting the number of vehicles and recording their types and speeds.	It has high accuracy and is quick to install. Also, it has low power consumption.	The initial cost of passive magnetic sensors are high. It has minor disruption to traffic during installation.	[62, 109]
Ultrasonic and passive acoustic sensors	These devices apply sound waves to detect vehicles by estimating the time that the signal returns to the device. The sensors are placed alongside the road and can be used to collect the number of vehicles and their speed.	It is easy for traffic flow observation.	The sensitivity of the sensor will be affected by the acoustic energy.	[4, 28, 98]
Video cameras	They record data including vehicle numbers, type, and speed using artificial intelligence techniques of image recognition.	It can provide more traffic information, not limited to the traffic flow and travel speed.	The cost of camera equipment is high.	[82, 90, 128]

Table 1.2 : Introduction of widely used non-stationary traffic detectors

Traffic detector	Principle	Advantage	Limitation	Typical Research
GPS	Recently, GPS has been widely used in the literature to validate the effectiveness of the proposed data-driven method. They are usually equipped in particular vehicles such as taxis, public transportations, and float vehicles. The precision of GPS is relatively high, typically less than 30m.	It has low cost, especially with the advanced technique of smart phone. Besides, it can be applied in any weather.	The signal of GPS is easily obstructed by the obstacles like buildings. Thus, it does not provide a precise location when vehicles are travelling in the city. In addition, GPS faces the problem of data sparsity caused by the low frequency of data sampling and a few available samples.	[80, 93, 105, 111]
Cellular-based systems	With the widespread use of smartphones, the position of vehicles can be located by tracing the smartphones equipped in the vehicles.	It is easy for traffic administrators to choose the positioning points of interest.	The cost is high. In addition, security and privacy are serious problems that should be overcome.	[12, 43]

road density, headway, and occupancy. The data is transmitted to the traffic control center and then stored, processed, and analyzed by transportation departments.

1.1.2 Advanced Techniques of Big Traffic Data Analytics

In [9], Biuk-Aghai indicated that data analytics mainly referred to the application of methods with respect to data storage and management, data preprocessing, and data analysis.

- **Data storage and management**

  There is a large amount of traffic data collected by the traffic detectors. In this case, advanced techniques of big data storage and management are usually based on Hadoop and related technologies, such as Hadoop distributed file system, HBase distributed database, and Hadoop MapReduce.

- **Data processing**

  We classify the data preprocessing operations into three categories, including data cleansing, data reduction, and data normalization. Data cleansing focuses on detecting, correcting or removing corrupt or inaccurate data from data set, and then replacing, modifying, or deleting the dirty or coarse data [75]. Data reduction is applied where the goal is to aggregate or amalgamate the information contained in large data sets into manageable (smaller) information nuggets [64]. Data normalization is a process where data attributes within a data model are organized to increase the cohesion of entity types.

- **Data analysis**

  The techniques of data analysis are mainly originated from the theories in economics and computer science. Time series methods are the most popular techniques in the field of economics and widely applied to big data analysis in

ITS such as Autoregressive Integrated Moving Average (ARIMA) model. Machine learning, as an advanced technique in the field of computer science, becomes more and more popular in recent years because of the rapid development of image processing and natural language processing, such as (un)supervised learning, reinforcement learning, and deep learning. With few variation, these methods can be also applied for traffic prediction and estimation [65, 70, 112].

1.1.3 Application of Big Traffic Data

Big data provides technical support for the development and applications of ITS, which improves the efficiency of ITS, reducing the costs, and having the benefit for the public of convenient and safe travel. Based on the survey made by Li et al. [134], big data transportation applications can be broadly divided into six categories, which are illustrated as follows:

- **Traffic accident analysis**

  According to the global status report in 2015 [67], around 1.2 million people are killed and 50 million injured from traffic accidents every year. With big traffic data analysis, traffic authority can make policies to prevent and reduce the occurrence of traffic accidents. To this end, in past years, many studies focus on using big data analytics in traffic accidents analysis.

- **Traffic state estimation**

  Traffic state analysis describes the process of the inference with respect to traffic state variables. These variables mainly include traffic flow, travel time and travel speed. With the knowledge of the traffic state, it is easy for traffic control and operations [83]. Thus, it attracts a lot of research attention in past decades like traffic flow prediction and travel time estimation.

- **Travel route planning**

With the widespread use of smartphones, a lot of applications have been proposed to enhance the travel experience of passengers like Google maps, Uber, and Didi. Google maps provides passengers with real-time public transport journey planning, enabling travelers to plan their trips moving from trains to buses and other transportation modes like taxis or bicycles. Uber and Didi provide the information about waiting passengers and vacant taxis which have the benefit of reducing the waiting time of both taxi drivers and passengers.

- **Public transportation services planning**

  With the big data of public transportation, transportation administrators can understand the patterns of passengers, which can be used to inform decisions of transportation operators about the planning of the services.

- **Rail transportation management and control** Rail transportation systems are closed systems where the data includes the speed, the position, and the departure and arrival time of the train. To improve the efficiency of rail transportation system operation and to help rail transport operators to control trains more closely, a lot of research is carried out to schedule the trains with advanced IT technology.

- Asset maintenance: asset management is a strategic and systematic process of maintaining, upgrading, and operating physical assets throughout their life cycle, such as roads, rail, and other traffic facilities [30]. By identifying problems with big data analysis, it is easy for traffic administrators to identify the issues quickly and accurately, and to carry out effective measures to maintain the asset at low cost.

1.2 Research Motivation

1.2.1 Traffic Flow Prediction

Traffic flow is an important parameter in the application of traffic state analysis. Accurate short-term traffic flow prediction can benefit both road users and traffic management authorities. On the one hand, road users can use traffic prediction to make better travel decisions, choose a faster route to reach the destination, and reduce fuel costs. On the other hand, traffic management authorities can utilize traffic prediction to improve traffic operation efficiency and apply more effective traffic control strategies to alleviate traffic congestion and improve the efficiency of road networks [1, 34, 44, 54, 55, 113].

Existing work for short-term traffic prediction suffers from a number of shortcomings. First, the accuracy of a prediction model heavily depends on the traffic flow data which is spatially and temporally correlated [59]. It is challenging for the prediction model to take full account of the intricate spatiotemporal correlation. Second, the spatiotemporal correlation between traffic at different observation points is not stationary but varies with time of the day [24]. To this end, multiple prediction models corresponding to different times of a day have been constructed to suit time-varying spatiotemporal traffic correlations [2, 18]. Third, many approaches adopt a black-box approach to traffic prediction, e.g., principal component analysis based techniques or neural network-based techniques. The parameters of the developed traffic prediction models lack physically intuitive explanations. As a consequence, it becomes very difficult, if possible, for traffic operators to adjust the model parameters to suit changing road topology and traffic conditions. Lastly, in two-dimensional road networks like urban road networks, the estimation of time-varying spatiotemporal correlation, which forms the basis of traffic prediction, becomes more intricate since the spatiotemporal correlation is also strongly affected by the trip distribution

and road topology.

1.2.2 Travel Time Distribution

Travel time is another traffic parameter that plays an important role in measuring traffic states of the road networks. In most studies, travel times are estimated at the level of link or path, where a link is usually defined as a oneway road segment without any road intersections inside, while a path is composed of a sequence of links [46, 96]. In fact, link travel time estimation delivers more benefits to both travelers and traffic administrators. First, it allows travelers to make optimal route choice to minimize their overall travel times. Second, traffic administrators can accurately locate where congestion happens, and carry out effective traffic management to improve road network performance accordingly.

Recently, travel time distribution (TTD) estimation has attracted considerable research attention. Unlike mean travel time estimation where travel time is estimated as a deterministic variable [71, 76, 95, 96], TTD estimation assumes travel time to be a random variable, which addresses the intuition that travel time is time-varying due to the heterogeneous and dynamic nature of traffic [63, 71, 129]. Moreover, the knowledge of the moments (mean and variance) obtained from a probability distribution can be used as the indicators to analyze travel time reliability [118].

Existing methods to estimate link TTDs suffer from the following shortcomings. First, many methods are based on parametric models like Gaussian distribution or log-normal distribution [103, 110]. These models are easy for mathematical analysis, yet unable to capture all of interesting dynamics of travel times that vary with the change of road conditions [76, 78, 95]. Second, to estimate the model parameters such as the means and variances in a Gaussian distribution, it is necessary to guarantee that there is sufficient travel time data in the target links [63]. Unfortunately,

this condition cannot be satisfied in an urban road network involving thousands of links, that are impractical to be fully covered by any type of data detector, e.g., GPS or traffic camera. Third, given the links where there is no observation, the travel times of these links are usually estimated based on contexts learned from their spatially and temporally correlated neighbors. However, the spatiotemporal correlation varies with the time of day [24]. Therefore, not only is it hard to guarantee the estimation accuracy of link travel times, but also a large amount of computing resources are consumed on data modeling [105].

1.2.3 Taxi Recommendation System

Taxi, as an important transportation mode in the metropolitan area, delivers a lot of convenience to our daily life. Unfortunately, much fuel and time is wasted when the taxis drivers are looking for the passengers [47, 61, 125]. To alleviate this problem, several efforts and studies have been made in time past to assist the taxi drivers to find passengers in time. Particularly, some of them have been successfully applied in the real world such as Uber and Didi.

Currently, noticeable research effort has been dedicated on taxi route recommendation [104, 126]. It aims to recommend a taxi driver with a hotspot, or the route to a hotspot, toward which the taxi driver is more likely to pick up a passenger. Therefore, the idle time of taxis and the waiting time of passengers can be both reduced. Compared with taxi dispatch, taxi route recommendation is more challenging since there is no real-time information with respect to taxi demand, as well as passengers' destinations.

Indeed, the strategies developed in the existing studies show good performance for taxi route recommendation. Unfortunately, they still suffer from some shortcomings. First, recent research primarily focus on route recommendation for a single taxi [39, 102, 125, 133]. Very few studies take account of the potential competition and

collaboration between taxis. As a result, taxis may travel towards the same hotspot according to the similar recommendation. On the one hand, taxi demand surplus may happen in a hotspot, causing some taxi drivers to not pick up any passengers. On the other hand, congestion easily happens in a hotspot due to the increase of traffic [74]. Second, existing methods are centralized approaches. Recommended routes for all the taxis in the urban area is analogous to a resource allocation problem with NP-hardness. Thus, it will consume a large amount of computational resources and take a lot of running time when the taxi management center exploits a centralized strategy to make the taxi route recommendation from a global perspective. Third, the techniques are mainly time series based methods, statistical methods and deep learning methods [61, 117, 119] where the input is usually Euclidean data like text or images. Euclidean data has a limitation on representing the spatial structure of the road network, which has the potential of analyzing the spatial correlation among the taxi demands. Some methods capture such spatial correlation through black-box approaches like convolutional neural network (CNN) and recursive neural network (RNN), while multiple hidden layers should be constructed and a great number of parameters need to be estimated.

1.3 Research Objectives and Contributions

From the aforementioned research background and motivation, this thesis focuses on the following research problems: 1) short-term traffic flow prediction, 2) link travel time distribution estimation, and 3) taxi cruising route recommendation. In the following section, we will elaborate on the detailed research problems and the corresponding contributions:

For the first research problem, we design a unified spatiotemporal model based on Space-Time ARIMA (STARIMA) which captures the intricate spatiotemporal correlation structure between road traffic and hence can potentially deliver more

accurate traffic flow prediction. Furthermore, parameters of the developed predictor have physically intuitive meanings, which make the model readily amendable to suit changing road topology and traffic conditions. Specifically, the main contributions regarding this research work are summarized as follows:

- A physically intuitive approach to traffic prediction is developed that captures the time-varying spatiotemporal correlation between traffic at different measurement points. Distinctly different from previous black-box approaches to road traffic modeling and prediction, parameters of the proposed approach have physically intuitive meanings which make them readily amendable to suit changing road and traffic conditions.

- Unlike some existing techniques which capture the variation of spatiotemporal correlation by a complete re-design and calibration of the model, the proposed approach uses a unified model which explicitly incorporates the impact of those physical factors affecting the variation of spatiotemporal correlation into the model parameters.

- Experiments using real traffic traces are conducted, which demonstrate that the proposed approach has superior accuracy compared with the STARIMA and the back propagation neural network (BPNN) based approaches, and is only marginally inferior to that obtained by constructing multiple STARIMA models for different time periods within the day, however with a much reduced computational complexity.

To address the challenges in the second research problem, we develop a non-parametric model for link TTD estimation, namely kernel density estimator (KDE). The model parameters are estimated using the data collected at or near the road intersections. The main contributions are briefly summarized as follows:

- With the proposed KDE based model, we are able to capture the dynamics of link travel times that vary with the change of road conditions. The model parameters are estimated with the proposed C-shortest path algorithm, K-means based algorithm, as well as the expectation maximization (EM) algorithm.

- We analyze the performance bottleneck of the proposed parameter estimation algorithms. To reduce the complexity and guarantee the estimation accuracy, we propose a Q-opt algorithm and an X-means based algorithm.

- We validate the proposed method based on a dataset including over 3.0e+07 GPS trajectories collected by the taxicabs in Xi'an, China. The experimental results show that the TTDs estimated using our proposed model are in excellent agreement with empirical distribution, provided that only $\sim 70\%$ of the intersections are equipped with traffic detectors.

To cope with the third research problem considered in this thesis, we aim at minimizing the number of vacant taxis to gain individual revenue and maximizing global revenue by considering collaboration and competition between taxis. The solution is obtained based on the predicted taxi demand using the proposed graph neural network (GNN) based method and the distributed algorithm based on Lagrange dual decomposition. The main contributions are summarized as follows:

- The graph-structured data in our proposed taxi demand predictor is based on the disjoint partition of the urban road network using a joint Speaker-Listener Label Propagation and GirvanNewman algorithm (SLP-GN) by considering road topology and geographical distribution of the position of interests (POIs).

- The taxi demand predictor takes account of the impact of multiscale features (hour and day) of taxi demand variation and spatiotemporal correlation between taxi demands in different pick-up zones. Experimental results using real

trajectory data collected from taxis in Xi'an, China show that the predictor has a better performance than its counterparts.

- The proposed distributed algorithm integrates an adaptive weighted-sum strategy to find the Pareto optimal solution to the bi-objective based problem. It has less complexity than widespread used centralized methods. The high efficiency enables the proposed distributed algorithm to be applied to solving real-time taxi dispatch problem.

- The simulation results show that the solution obtained based on our proposed scheme is better than existing single-objective based optimal solutions from the perspectives of reducing vacant taxis and gaining global revenue of taxis.

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 presents a survey of related works, including traffic flow prediction, travel time estimation, and the taxi recommendation system. Chapter 3 proposes a novel STARIMA based model for short-term traffic flow prediction with consideration of spatial-temporal correlation between traffic. Chapter 4 develops an estimation model of link travel time distribution with limited traffic detectors configured at or near the road intersections. Chapter 5 proposes a distributed scheme for taxi cruising route recommendation. There are three components in the recommendation system, namely community detection, taxi demand prediction, and the solution based on Lagrange dual decomposition. Chapter 6 presents a summary of the thesis contents and recommends future works.

Chapter 2

Literature Survey

This chapter is dedicated to reviewing related works to this thesis, including works on cooperative vehicular networks, capacity of vehicular networks and security of vehicular networks.

2.1 Short-term Traffic Flow Prediction

Depending on the traffic information employed for prediction, short-term traffic prediction models can also be classified into: (i) temporal models which predict future traffic at a particular location of interest using historical (temporal) traffic data at the same location [8, 108, 116], (ii) spatio-temporal models which explore both historical traffic information and traffic information of spatially close measurement points for prediction [15, 45, 54].

Temporal models have been extensively applied in the past two decades. Particularly, time series based methods such as the ARIMA model and its variants have attracted significant attention [1, 99, 108]. Mascha et al. proposed a Kohonen ARIMA (KARIMA) model, which applies Kohonen self-organizing map technique to classify the input data into a set of clusters, and then establishes an individually tuned ARIMA model for each cluster [99]. Williams et al. [108] developed a seasonal ARIMA (SARIMA) model, which tries to identify seasonal patterns in the traffic to capture the cyclical variation of traffic states, such as peak and off-peak hours in each work day. In another work, Afshin et al. [1] used the SARIMA model to obtain accurate short-term prediction with limited input data. To capture

the stochastic and nonlinear characteristics of historical traffic data in a temporal model, techniques from areas such as machine learning, economics, and stochastic analysis are also employed by researchers for traffic flow prediction. Some examples include Artificial Neural Network (ANN) [45, 89], Bayesian Network (BN) [14] and Support Vector Regression (SVR) [40]. However, spatial traffic correlation that can potentially be explored to improve the prediction accuracy was not considered in the aforementioned research.

To overcome the above shortcomings, spatio-temporal models have emerged as an efficient way to improve the prediction accuracy. Williams [107] developed a multivariate ARIMA model, denoted by ARIMAX (ARIMA with **ex**ogenous variables), which uses exogenous variables to capture the influence of upstream flows on downstream flows. An extension based on the ARIMAX model was developed by Stathopoulos and Karlaftis [91] by setting up various ARIMAX models for different time periods of the day. Xia et al. [113] proposed a spatio-temporal weighted K Nearest Neighbor (KNN) model, named STW-KNN, which predicts the traffic flow of a road by finding the most correlated flow from historical records at K adjacent up/downstream roads. The novelty of their research lies in the adoption of a state vector to describe the traffic conditions and a suitable distance metric to determine the proximity and correlation of traffic flows at different roads. In [94], Sun et al. modeled the road network as a Bayesian network where a road is represented as a node and the causal relation between two adjacent roads is represented as an edge. The joint probability distribution between the nodes with known data and the ones to be predicted was described by a Gaussian mixture model (GMM) where the parameters are estimated using the competitive expectation maximization algorithm. Bayesian network is also applied in Horvitz et al.'s work [38] which modeled traffic flow in the road, as well as the factors (e.g., incident, major events, weather) potentially affecting the variation of traffic flow as the nodes in the Bayesian net-

work. To find the causal relation between nodes, a heuristic search together with a Bayesian scoring criterion to guide the search was performed over the models. Lv et al. [55] considered the traffic data as variables in the space-time cube. The generic traffic flow features embedded in these input variables are learned by a stacked auto-encoder model, a kind of neural networks. The model is trained in a greedy layerwise fashion and then used for forecasting. Deep learning was also usedin [70] where Polson and Sokolov applied l_1-regularization technique to identify the spatio-temporal patterns. The experimental results showed that the preidctor was able to provide precise short-term traffic flow predictions even in the case that traffic flow regime changed drastically. Mitrovic et al. [60] used a singular value decomposition (SVD) based technique to construct a relationship matrix with which the traffic data of a few selected roads is able to map to that of the whole network. The traffic flows of the selected roads are then predicted by the SVR models and extrapolated to the whole network using the aforementioned relationship matrix.

Another major class of spatio-temporal models is the STARIMA based methods. In the STARIMA, a spatial weight matrix W is introduced that comprises two components: a spatial adjacency structure and a spatial weighting structure [15, 59]. As for the spatial adjacency, it reflects first-order spatial relations between all observations where two directly adjacent observations are termed as first-order spatial neighbors. For the spatial weight, it is the element of W that expresses the spatial correlation between two first-order neighbors. The parameters in the STARIMA model are (p_λ, d, q_m) where p and q are time lags for the STAR and the STMA models respectively, d is the degree of differencing, λ and m are the spatial orders for the STAR and the STMA models respectively. The improvements in the performance of a STARIMA model are primarily shown in the aspect of capturing the temporal variation of spatio-temporal correlation. A common method is to re-estimate the parameters of the STARIMA model in each traffic state of the day

to better capture the traffic similarity in the same state. For example, Min and Wynter [59] redefined a spatial order as an ordering with respect to the Euclidean distance traveled by vehicles within a unit time interval. As travel speed varies temporally, the spatial weight matrix is re-evaluated in different time periods of the day. Similarly, Cheng et al. [15] transformed the static spatial weight matrix into a dynamic one by defining the spatial weight as a function of the time-varying speed between two neighboring locations. Unfortunately, with the rapid variation of traffic conditions, this causes a large increase in the number of estimated parameters and an explosive growth of computational time. To improve the efficiency of estimating the parameters in multiple STARIMA models corresponding to different times of the day, Salamanis et al. [79] only employed a prescribed number of spatially correlated neighbors of a road of interest. They analyzed the degree of the spatio-temporal correlation between the traffic from different measurement points using a Pearson product–moment correlation-coefficient-based metric, which is based on the cross correlation function. One of our work [24] proposed a convenient technique to adjust the lags of the STARIMA model dynamically to suit different traffic states, which was validated using measured traffic data on a highway.

To apply the approach developed in [24] to an intricate two-dimensional road network, a number of challenges need to be conquered, including the explicit consideration of the road topology and trip distribution in traffic prediction. As for the road topology, most studies use graph-theoretic techniques to transform a road network into a mathematical model convenient for subsequent analysis. Kelly in [42] modeled the road network by an incidence matrix. Each column in the matrix corresponds to a road and each row corresponds to a measurement point in a road. The column for a road comprises entries of 0s or 1s with 1 indicating a particular measurement point is on a particular road and 0 otherwise. The 1s in a row suggest which roads pass through that measurement point. However, the dimension

of an incidence matrix quickly explodes for even a moderate number of roads and measurement points. To overcome the scalability problem, Salamanis et al. in [79] represented a road network by an adjacency matrix where each column and each row represented a road. If two roads are adjacent, the corresponding entry in the adjacency matrix is 1; otherwise, the entry becomes 0. It is worth noting that all aforementioned methods modeled the road network as an undirected graph, that is, prediction must be executed before specifying a particular traffic direction. Unlike existing graph-theoretic techniques, we employ a digraph to model the road network which can better capture directionality of road traffic flows.

In the literature, the spatial pattern of traffic between origins and destinations is usually expressed by a trip distribution matrix based on the undirected graph model of traffic network and widely used in the traffic state estimation [5], traffic flow prediction [1] or traffic flow demand estimation [22] and so forth. To extend the trip distribution matrix to the digraph model, we propose the concepts of turning rate and traffic transition probability (TTP) which are capable of accurately capturing the traffic distribution among roads with road intersections. To estimate turning rate or TTP, we apply the gravity model based method where not only traffic data, but also the spatial separation between two locations is considered [131]. As the gravity model merely requires the traffic information at the origin and the destination, the adverse impact of missing traffic measurements on some roads along the paths between the origin and the destination can be omitted. Indeed, in the real life, it is economically prohibitive to deploy traffic detectors across the whole road network.

2.2 Link Travel Time Estimation

Research on link travel time estimation are mainly divided into two categories: 1) mean travel time estimation, and 2) TTD estimation. In the remainder of this section, we first introduce the work in terms of mean travel time estimation according

to the utilization of traffic detectors. After that, the research with respect to TTD estimation are summarized and analyzed according to the application of probabilistic models.

Data driven methods are well studied in the mean travel time estimation because of the coming of big data era for transportation. As mentioned in Introduction, these traffic data are collected by different types of detectors, mainly including GPS, Bluetooth device, loop detector and traffic camera. A significant part of GPS based methods is devoted to solve map matching and data sparsity problems. Map matching aims to calibrate the GPS coordinates that did not fall into the roads where the vehicles were traveling in. The corresponding approaches are on the basis of geometric, topological, probabilistic, and artificial intelligence. The details of these approaches have been introduced in Sanaullah et al.'s work [80]. Data sparsity is mainly caused by the low sampling frequency and limited number of data. The solve this problem, the distinguished work was done by Wang et al. [105] and Tang et al. [96]. They both introduced the tensor, a technique utilized in the deep learning, to model travel times on different links as multi-linear manner geometric vectors. As the neighboring links are spatially correlated, the tensor without observed data were estimated based on the geospatial, temporal and historical contexts learned from the neighboring tensors.

Bluetooth device is an alternative to provide traffic data for mean travel time estimation. The Bluetooth device in each vehicle has the unique Media Access Control address (MAC address). The traffic information of these vehicles will be captured by the Bluetooth Traffic Monitoring Systems (BTMSs) installed in the roads. Bluetooth based methods focus on solving the following two issues: i) the measurement reliability produced by BTMSs, e.g., transmission power, and ii) the probability of detecting the same vehicle by two successive Bluetooth detectors. To get rid of these problems, Ashish, Ming and Edward [6] calibrated Bluetooth data

with the aid of loop detectors. Vinagre et al. [21] introduced a series of weights to adjust the travel times estimated from Bluetooth data. The weights were predefined according to different traffic patterns such as free flow or congestion.

Loop detector is also a widely used detector in mean travel time estimation. Unlike GPS and Bluetooth, it cannot identify the vehicles. As a result, related research are usually on the basis of traffic flow theory. In other words, travel times were inferred from traffic flow and travel speed [49, 122]. For instance, Li et al. [49] designed an a temporal-spatial queuing model with consideration of travel speed, headway time series and travel times. Yi and Williams [122] proposed a dynamic Nam-Drew model to estimate travel times under traffic conditions of free flow and congestion.

Traffic cameras collect data through videos or images. With the rapid development of the techniques in the realm of artificial intelligence [86, 88], the improvement of vehicle recognition accuracy enable the traffic camera data to become more reliable. Unfortunately, it is impractical to monitor every link in the whole urban network by traffic cameras. To address this problem, Yeon et al. [121] developed a Discrete Time Markov Chains (DTMC) model with consideration of different road conditions like congestion and free flow. Rahmani et al.Rahmani et al. [77] proposed a method extended from kernel-based estimation by means of both traffic camera data and GPS data.

In recent years, growing interest is motivating a shift toward estimating TTD. The corresponding work usually assume travel times follow either Gaussian distribution or log-normal distribution. Specially, Li et al. [50] indicated Gaussian was appropriate to model travel time in the presence of free flow, small time interval (e.g., 5 minutes), whereas log-normal was appropriate to model travel time in the presence of congestion, large time interval (half an hour). Other probabilistic models

were also used to model travel times such as Weibull distribution [3] and Burr distribution [33]. To improve the estimation accuracy, Pu et al. [72] used the log-normal model with consideration of the inter dependencies between the reliability measures of travel times such as standard deviation, coefficient of variation and frequency of congestion. Moylan and Rashidi [63] constructed multiple hazard-based models under different road conditions leveraging on the factors affecting the variation of road conditions such as the weather, the wind speed were modeled as explanatory variables. Prokhorchuk et al. [71] proposed a Gaussian copula graphical model to transform the non-Gaussian characteristics of travel times into Gaussian. Yang et al. [118] developed a Gaussian mixture model by considering the delay in the signalized intersections. From above literature, we can observe that these methods are parametric model based. As the structure (the number of parameters) of a parametric model is fixed, it is difficult for them to capture all of interesting dynamic of travel times varying with the change of road conditions [24].

A common weakness in the existing research on TTD estimation is that they seldom consider impact with respect to the limited coverage of traffic detectors. This is because the proposed estimators are generally implemented in the typical study sites like the major roads in the urban city [63]. The traffic detectors deployed in these study sites are dense. Thus, there are always sufficient observations. However, consider the whole urban network, a traffic detector, e.g., traffic camera, is far away from another. It leads to a problem that the traffic states in the links between two traffic detectors are unobservable. To solve this problem, techniques of network tomography are the candidates. More concretely, network tomography uses the information derived from end-to-end (E2E) measurements to explore the internal characteristics of an internet network, e.g., the packet transmission delay. In the context of traffic network, the travel times detected by the two traffic detectors can be viewed as the E2E measurements. Thus, the TTDs of the links between the

two traffic detectors can be inferred from the observations. Motivated by this idea, Zhang et al.'s [129] provided a traffic camera deployment strategy with which the accuracy of mean travel time estimation was improved, and meanwhile, the overall deployment cost on traffic cameras was minimized.

Different from Zhang et al.'s work, we focus on TTD estimation. Although similar problem like the estimation of link delay distribution has been researched in the network tomography, the techniques cannot be directly used in our work since most of them are still based on parametric models such as Gaussian or exponential distributions. Note that bin size model is a kind of non-parametric model [25, 97] used in the network tomography, however, it can vary wildly with the different configuration of bins, especially with relatively small number of data. Therefore, in this paper, we use kernel density estimator which provides similar distribution even with varying bandwidth and/or kernel type.

2.3 Taxi Recommendation System

2.3.1 Taxi Route Recommendataion

In the existing literature, a hotspot or the route to this hotsport recommended to a taxi should satisfy the objectives such as: i) miminizing the fuel cost [74], ii) maximizing the taxi revenue [47, 133], or iii) maximizing the probability to find a passenger [126, 127]. To this end, the corresponding techniques are summarized as follows.

Yuan et al. [127] extracted the hotspots based on the knowledge of passengers' mobility patterns and taxi drivers' pick-up/drop-off behaviors learned from the historical taxi trajectories by means of a probabilistic model. A similar model was also used to learn the behaviors of taxis in Qu et al.'s work [74] who employed a brute-force strategy to generate an optimal driving route for a taxi. Huang et al.

[39] searched the candidate hotspots for a taxi with the backward incremental sequence generation algorithm. Besides, an incremental pruning policy was applied to reduce the searching space. After that, a batch pruning algorithm was used to find the optimial solution. In [102], Verma et al. modeled the action of a taxi traveling from one position to another as the state transition where a state is defined as a taxi's location in a time interval. Thus, a recommended route for a taxi is a sequence of states and predicted by a reinforcement learning based method. Moreover, a dynamic abstraction mechanism was used to improve the basic learning mechanism. Both Zhou et al. [133] and Yu et al. [125] modeled the problem as a Markov Decision Process (MDP). The difference between these two studies relies on the following aspects: The former work applied the probabilistic model to estimate the probabilities that a taxi traveled to different pick-up locations and destinations. After that, a rolling horizon configuration strategy was proposed to get the optimal solution. The latter work modeled the pick-up location and taxi destination based on homogeneous poisson process (HPP). The solution was then obtained with a value iteration algorithm. A novel method was proposed by Lai et al. [47] who applied Coulomb's law into route recommendation. In this study, the taxis and passengers are modeled as positive and negative charges. The forces between the charges are regarded as the attractiveness between taxis and passengers. An optimal route was the one from the current location of a taxi to a passenger's location who had the strongest attractiveness to such taxi.

Applying the above approaches into our work will face a challenge, that is, the increment of computational complexity due to the consideration of competition and collaboration between taxis. In this case, a distributed framework is more feasible since it is more suitable for the transportation systems, which are usually geographically distributed in dynamic changing environments. Although there is no distributed solution for taxi recommender problem, it has been developed for other

similar problems such as dynamic routing [106], congestion management [53], and intelligent traffic control [32]. Particularly, the multi-agent system is one of the effective techniques for distributed modeling and simulation [13]. However, a large number of agents and the complicated relationship between agents will increase the complexity of modeling and distributed algorithm [10]. Another approach is to apply numerical methods like mixed integer programming (MIP) [10]. It enables the closed-form solution. Therefore, in this paper, we apply MIP to model our multiple-objective problem and solve the problem with distributed Lagrange dual decomposition based algorithm.

2.3.2 Taxi Demand and Destination Prediction

With advancements in artificial intelligence, applying deep learning techniques like CNN, RNN and their variants into taxi demand and destination prediction has shown an increasing trend over the time period. In [20], RNN with shallow structure was developed to predict taxi destination. Ke et al. [41] forecasted taxi demand with a Fusion Cnvolutional Long short-term memory Network (FCL-Net). It captured the spatial dependencies, temporal dependencies, and exogenous dependencies between historical taxi trajectories. In [117], Xu et al. applied a sequence learning model, namely long short term memory (LSTM) to predict taxi demand where LSTM had ability of storing relation between historical and future traffic information. In [130], Zhang et al. proposed a novel method by incorporating SVR and ensemble learning approach to predict taxi destination.

Another category of techniques are based on the time series models. For instance, Matias et al. [61] proposed a predictor combining two time-series forecasting techniques, respectively time-varying Poisson model and ARIMA model. Different time-series models were also employed in Davis et al.'s work [19] including seasonal and trend decomposition using Loess (STL), HW (Holt Winters) model and ARIMA

model. Zhao et al. [132] came up three predictors based on the the Markov process, the Lempel-Ziv-Welch text encoding algorithm, and the Neural Network by considering temporal correlation of human mobility to measure the demand uncertainty.

Aside from the aforementioned methods, the models used for traffic flow, travel time and congestion prediction, e.g., STARIMA [23], genetic algorithms [54]. etc, also have the potential of forecasting taxi demand and destination. Note that the inputs of above predictors are mainly time series, also known as Euclidean data. However, graph representation of traffic is more reasonable since road network is usually modeled as a graph in the research of transportation [96, 129]. Motivated by this observation, GNN has gained increasing popularity since it has the ability of modeling the dependencies between nodes in a graph, and thus enables the breakthrough in the research area related to graph analysis.

Recently, a few studies applied GNN into traffic prediction. For instance, Yu et al. [124] predicted the future vital indicators (e.g., speed and volume) with a spatial-temporal graph convolutional neural network (STGCNN). More precisely, STGCNN stacks multiple spatial-temporal conv-blocks where each conv-block contains the two-layer structure including a GNN layer and a CNN layer. Traffic data in the neighboring roads are usually spatiotemporally correlated. The variation of spatiotemporal correlation is periodical in a day or in a week, so that the features we learn from the data within a particular time period can also be applied to other time periods. With convolution operation, the aforementioned features can be extracted and filtered from input training samples, and can be used at all time periods. In general, as the depth of the neural network model increases, the complexity of features learnt by convolution layers increases. However, to make a tradeoff, in the paper, the authors only considered two layers in each block. In [51], Li et al. forecasted traffic flow with a proposed diffusion convolutional recurrent neural network (DCRNN) that captured the spatial correlation using bidirectional random walks

on the graph, and the temporal dependency using the encoder-decoder architecture with scheduled sampling. A distinguished work was done by Xu et al. [31] who proposed a spatio-temporal multi-GCN (STM-GCN). They first modeled differnt types of spatial correlations (neighborhood, functional similarity and transportation connectivity) among regions by multiple graphs. Then spatial correlation was extracted using multi-graph convolution. To capture the temporal correlation, they applied a recurrent neural network augmented by a contextual-aware gating mechanism, that is, assigning different weights to historical observations.

2.4 Summary

In this chapter, the literature is reviewed with respect to short-term traffic flow prediction, link travel time estimation and taxi recommendation. More precisely, typical parametric and non-parametric models for traffic flow prediction are described, followed by the approaches to link travel time estimation including mean travel time and travel time distribution estimation. Finally, to illustrate existing methods regarding taxi recommendation, studies on taxi demand and destination prediction are elaborated.

Chapter 3

Unified Spatio-temporal Model for Short-term Traffic Flow Prediction

In this chapter, we develop a traffic flow predictor based on STARIMA model in which we consider spatiotemporal correlaiton between traffic at different measurement points. As the spatiotemporal correlation varies with the time, we analyze the impact of physical factors such as road network topology, time-varying speed, and time-varying trip distribution on the time-varying spatiotemporal correlation. After that, we incorporate above physical factors into a series of parameters, and hence, these parameters are relatively easy to control and adjust when road and traffic conditions change. Due to the fact that there is no need to re-estimate all the parameters in the predicictor, thereby it greatly reduces the computational complexity. Experiments using two set of real traffic traces, respectively collected from one-dimensional highway and two-dimensional road network, demonstrate that the proposed approach has superior accuracy compared with the widely used STARIMA and BPNN based approaches, and is only marginally inferior to that obtained by constructing multiple STARIMA models for different time of the day, however with a much reduced computational and implementation complexity.

The rest of this chapter is organized as follows. In section 3.1, we first introduce typical STARIMA model, followed by the proposed unified STARIMA model. The methodologies of model parameters estimation are illustrated in Section 3.2. The performance of the proposed methods are evaluated in Section 3.3. Finally, Section 3.4 draws the summary.

3.1 Unified Spatio-temporal Model

In this section, we introduce the system model and the method of parameter estimation in the proposed model.

3.1.1 STARIMA Model

For completeness, we first introduce the STARIMA(p_λ, d, q_m) model which is defined as follows:

$$\begin{aligned}&(I - \sum_{k=1}^{p}\sum_{l=0}^{\lambda_k}\phi_{kl}W_l L^k)(1-L)^d Y(t)\\ =&(I - \sum_{k=1}^{q}\sum_{l=0}^{m_k}\theta_{kl}W_l L^k)\epsilon_t. \end{aligned} \tag{3.1}$$

In this chapter, $Y(t) = \{y_1(t), y_2(t), ..., y_N(t)\}$ is a $\mathcal{N} \times 1$ vector including the traffic flow from $\mathcal{N}$ links at$t$, L is the lag operator: $y_i(t-1) = Ly_i(t), i \in \mathcal{N}$, ϕ_{kl} and θ_{kl} are coefficients, W_l is the spatial weight matrix, and ϵ_t is white noise. There are three steps to set up a STARIMA model including 1) Model Identification, 2) Parameter Estimation and 3) Diagnostic Checking [69].

3.1.2 System Model

To better illustrate the establishment process of the unified spatiotemporal model, we model a road network as a digraph. We partition the road network into a set of *road segments*. Each road segment is a piece of road bounded by two road intersections and there is no intersection within a road segment. A very long road segment may be further partitioned into multiple smaller road segments. We call a particular travel direction of a road segment a *link*. Depending on whether the road is one-way or two-way, a road segment may be represented by one or two links [42]. Without losing generality, we further assume that there is at most one measurement point within a link. If there are multiple measurement points within a long road, this can be readily handled by dividing the long road into multiple road segments where each

segment contains up to one measurement point only. Drawing from graph theory, an arrangement of links can be modeled as a digraph $D = (V, E)$ with a set of V of vertices and a set E of arcs (or directed edges). The vertex set $V = \{V_1, V_2, ..., V_{\mathcal{N}}\}$ and $V_i \in V$ represents the i-th link or a particular point, e.g., a measurement point, if it exists, within the i-th link. There is an arc $e_{i,j} \triangleq (V_i, V_j)$, $e_{i,j} \in E$, going from V_i to V_j if there is traffic traveling *directly* from V_i to V_j. Based on the digraph model, a route from link i to link j is defined as a path from V_i to V_j, including a finite sequence of arcs connecting a sequence of vertices that are all distinct from one another. Moreover, the number of arcs is denoted by l, which is the *path length*. Since a vertex $V_i \in V$ has both incoming and outgoing arcs, the neighbors of V_i are classified into two categories. The first category is a set of vertices that are the links located upstream of link i. We denote it by V_i^{1-}. Correspondingly, the second category is a set of vertices including the neighbors of V_i that are the links located downstream of link i. We denote it by V_i^{1+}.

In the following, we first explore the spatiotemporal correlation between $V_i \in V$ and $V_j \in V_i^{1+}$. To begin with, we introduce the concept termed "turning rate" $\pi_{i,j}$ to represent the ratio of traffic at V_i and traveling to V_j. Then, we approximately estimate the incoming traffic at V_j from V_i by:

$$y_{i,j}(t) = \pi_{i,j} y_i(t - \tau_{i,j}), \tau_{i,j} \in \mathbb{Z}^+, \tag{3.2}$$

where $y_i(t - \tau_{i,j})$ represents the traffic at V_i at $t - \tau_{i,j}$. $\tau_{i,j}$ is the time-varying lag corresponding to the time required to travel from V_i to V_j because at that time lag, the (approximate same) set of vehicles $y_{i,j}(t)$ have reached V_j. Note that, the turning rate $\pi_{i,j}$ varies over the time of the day. In this paper, we assume that $\pi_{i,j}$ remains constant during a given time period of the day, e.g., peak or off-peak hours. The estimation of $\pi_{i,j}$ and $\tau_{i,j}$ will be discussed in next section. Utilizing the lag

operator L, Equation (3.2) can be rewritten as

$$y_{i,j}(t) = \pi_{i,j} L^{\tau_{i,j}} y_i(t), \tau_{i,j} \in \mathbb{Z}^+. \tag{3.3}$$

Based on (3.2), we obtain the traffic at V_j:

$$y_j(t) = \sum_{V_i \in V_j^{1-}} y_{i,j}(t). \tag{3.4}$$

Unfortunately, not every vertex in V_j^{1-} has measurement data available since in real applications many links may not be equipped with traffic detectors. Denoting the subset of vertices with measurement data in V_j^{1-} by $\widehat{V_j^{1-}}$, whereas the subset of vertices without measurement data by $\widetilde{V_j^{1-}}$. In this case, $y_j(t)$ can be expressed as the sum of the traffic coming from $\widehat{V_j^{1-}}$ and $\widetilde{V_j^{1-}}$:

$$y_j(t) = \sum_{V_{i_1} \in \widehat{V_j^{1-}}} y_{i_1,j}(t) + \sum_{V_{j_1} \in \widetilde{V_j^{1-}}} y_{j_1,j}(t). \tag{3.5}$$

In (3.5), the traffic from $\widehat{V_j^{1-}}$ can be calculated directly. As for the traffic from $\widetilde{V_j^{1-}}$, we should estimate it by considering the traffic upstream from the adjacent neighbors of V_{j_1}. Moreover, if there is still no measurement traffic upstream from the adjacent neighbors of V_{j_1}, we have to further consider the traffic upstream from the neighbors that are far away from V_{j_1}. For the sake of simplicity, we term $\widehat{V_j^{1-}}$as the first *in-level-available* vertices of V_j. Second *in-level-available* vertices of V_j, denoted by $\widehat{V_j^{2-}}$, and so on. Correspondingly, $\widetilde{V_j^{l-}}, l \geq 1$ are termed as the l-th *in-level-unavailable* vertices of V_j. To find $\widehat{V_j^{l-}}$ and $\widetilde{V_j^{l-}}$ in the general case, a BFS (breadth first search) based algorithm is designed and applied. We will present such algorithm in next section.

We use $P_{i_l,j}^l, V_{i_l} \in \widehat{V_j^{l-}}$, to denote a set of paths where each path $P_z \in P_{i_l,j}^l$ starts from V_{i_l} and ends at V_j via $l-1$ vertices respectively belong to $\widetilde{V_j^{1-}}, \widetilde{V_j^{2-}}, ..., \widetilde{V_j^{(l-1)-}}$. We use $y_{i_l,j}^l(t)$ to denote the traffic traveling from V_{i_l} to V_j along $\forall P_z \in P_{i_l,j}^l$. Besides,

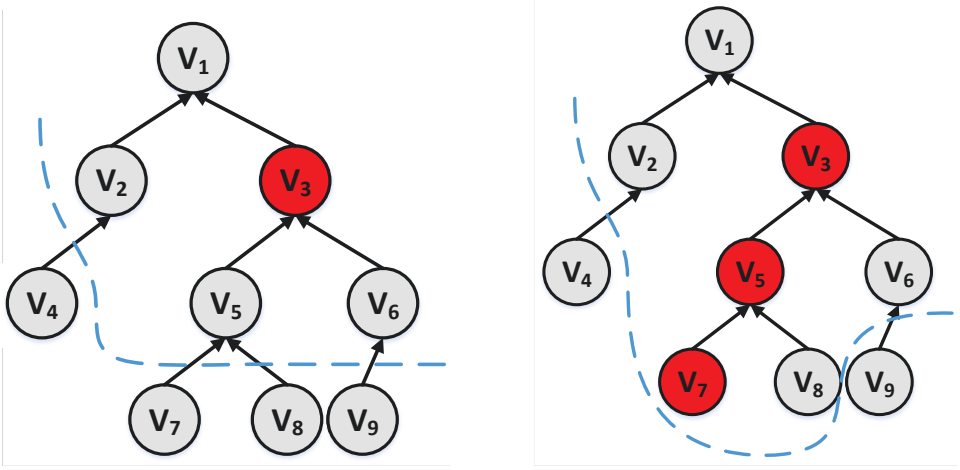
$y^l_{i_l,j}(t)$ is estimated by

$$\begin{aligned} y^l_{i_l,j}(t) &= \sum_{p_z \in P^l_{i_l,j}} \pi^{P_z}_{i_l,j} y_{i_l}(t - \tau^{P_z}_{i_l,j}) \\ &= \sum_{p_z \in P^l_{i_l,j}} \pi^{P_z}_{i_l,j} L^{\tau^{P_z}_{i_l,j}} y_{i_l}(t). \end{aligned} \tag{3.6}$$

Particularly, $\pi^{P_z}_{i_l,j}$ and $\tau^{P_z}_{i_l,j}$ are respectively the turning rate and time-varying lag between V_{i_l} and V_j upon path $P_z \in P^l_{i_l,j}$. Both $\pi^{P_z}_{i_l,j}$ and $\tau^{P_z}_{i_l,j}$ are estimated on the basis of $\pi_{i,j}$ and $\tau_{i,j}$. Suppose there is a λ_j satisfying $\widetilde{V_j^{\lambda_j-}} = \varnothing$. With (3.6), we can calculate $y_j(t)$ by

$$y_j(t) = \sum_{l=1}^{\lambda_j} \sum_{V_{i_l} \in \widehat{V_j^{l-}}} y^l_{i_l,j}(t). \tag{3.7}$$

Up to this point, we draw a clear and physically intuitive picture of the spatiotemporal correlation between any two links. To better illustrate above process, we give an artificial instance in Fig. 3.1a where the gray nodes are the vertices with measured data, whereas the red nodes are the vertices without measured data. In this instance, $\widehat{V_1^{1-}} = \{V_2\}$ and $\widetilde{V_1^{1-}} = \{V_3\}$. As there is no traffic measured at V_3, we should consider $\widehat{V_1^{2-}} = \{V_5, V_6\}$. Since $\widetilde{V_1^{2-}} = \varnothing$, we get $\lambda_1 = 2$. Finally, we calculate $y_1(t) = y^1_{2,1}(t) + y^2_{5,1}(t) + y^2_{6,1}(t)$. In (3.7), a big challenge we face is that in some situations, there is no such value of λ_j which satisfies $\widetilde{V_j^{\lambda_j-}} = \varnothing$. In other words, there are no enough detectors to provide sufficient data to calculate $y_j(t)$. Consider Fig.3.1b where the digraph structure of the road network topology is the same as the one in Fig.3.1a. However, not only V_3 but also V_5 and V_7 do not have measured data. In order to estimate the traffic at V_3, the BFS algorithm will be executed until the leaf node V_7 is achieved. As the traffic at V_7 can not be inferred from its child nodes, it is impossible to accurately estimate the traffic at V_5 and V_3. Furthermore, the traffic at V_1 can not be calculated via (3.7). To tackle this problem, we assume that a BFS algorithm terminates when there is $l = \lambda_j$ satisfying each node in $\widetilde{V_j^{\lambda_j-}}$ has no child. In this way, $y_j(t)$ consists of two parts. The first part is the traffic from measured links while the second part is the traffic from unmeasured links. Thus



(a) Traffic flow prediction for V_1 with enough traffic data

(b) Traffic flow prediction for V_1 without enough traffic data

Figure 3.1 : Traffic flow prediction for a vertex (link) in an artificial road network with consideration of the situations that there is (not) enough traffic data

(3.7) can be expressed as follows:

$$y_j(t) = \sum_{l=1}^{\lambda_j} \sum_{V_{i_l} \in \widehat{V_j^{l-}}} y_{i_l,j}^{l}(t) + \sum_{V_i \in \widetilde{V_j^{\lambda_j -}}} y_{i,j}^{\lambda_j}(t) \tag{3.8}$$

For simplicity, we use $\widehat{y_j(t)}$ and $\widetilde{y_j(t)}$ to represent the first and second part in (3.8) respectively. Based on (3.2), (3.3) and (3.8), $\widehat{y_j(t)}$ can be estimated by

$$\widehat{y_j(t)} = \sum_{l=1}^{\lambda_j} \sum_{V_{i_l} \in \widehat{V_j^{l-}}} \sum_{p_z \in P_{i_l,j}^{l}} \pi_{i_l,j} L^{\tau_{i_l,j}} y_{i_l}(t) \tag{3.9}$$

Assuming that there are $\hat{\mathcal{N}} \leq \mathcal{N}$ links in the road network with measured data. We then define two $\hat{\mathcal{N}} \times 1$ vectors $Y(t) = \{y_j(t) | j \in \hat{\mathcal{N}}\}'$ and $\widehat{Y(t)} = \{\widehat{y_j(t)} | j \in \hat{\mathcal{N}}\}'$. Then, (3.9) can be expressed as

$$\widehat{Y(t)} = \sum_{l=1}^{\lambda} \widehat{\phi}_l Y(t), \tag{3.10}$$

In (3.10), $\widehat{\phi}_l$ is a $\hat{\mathcal{N}} \times \hat{\mathcal{N}}$ matrix where the $(i,j)^{th}$ entry is $\sum_{p_z \in P_{i,j}^{l}} \pi_{i,j} L^{\tau_{i,j}}$ if $V_i \in \widehat{V_j^{l-}}$, Otherwise, the entry is equal to 0. Beside, we define λ as the maximal value of

$\lambda_j, j \in \hat{\mathcal{N}}$, mathematically, denoted as

$$\lambda = \underset{\forall j \in \hat{\mathcal{N}}}{max}\ \lambda_j. \tag{3.11}$$

With estimated results of π, τ and λ (using the methods in the next section), $\widetilde{y_j(t)}$ can be calculated by $y_j(t) - \widehat{y_j(t)}$. We define a $\hat{\mathcal{N}} \times 1$ vector $\widetilde{Y(t)} = \{\widetilde{y_j(t)} | j \in \hat{\mathcal{N}}\}$. Then we construct a STARIMA model for $\widetilde{Y(t)}$, formulated as follows:

$$\widetilde{Y(t)} = \sum_{k=1}^{p}\sum_{l=0}^{\lambda_k} \widetilde{\boldsymbol{\phi}_{kl}} \boldsymbol{W}_l L^k \widetilde{Y(t)} + \varepsilon_t - \sum_{k=1}^{q}\sum_{l=0}^{m_k} \widetilde{\boldsymbol{\theta}_{kl}} \boldsymbol{W}_l L^k \varepsilon_t. \tag{3.12}$$

Unlike original STARIMA model where l refers to the spatial order between two vertices, in (3.12), l is the path length. The $(i,j)^{th}$ entry of W_l is 1 if $V_i \in \widehat{V_j^{l-}}$. Otherwise the entry is 0. Eq. (3.12) can also denoted as

$$\begin{aligned} &(\boldsymbol{I} - \sum_{k=1}^{p}\sum_{l=0}^{\lambda_k} \widetilde{\boldsymbol{\phi}_{kl}} \boldsymbol{W}_l L^k)(1-L)^d \widetilde{Y(t)} \\ =&(\boldsymbol{I} - \sum_{k=1}^{q}\sum_{l=0}^{m_k} \widetilde{\boldsymbol{\theta}_{kl}} \boldsymbol{W}_l L^k)\varepsilon_t. \end{aligned} \tag{3.13}$$

According to (3.10), we have

$$\widetilde{Y(t)} = Y(t) - \widehat{Y(t)} = (\boldsymbol{I} - \sum_{l=1}^{\lambda} \widehat{\boldsymbol{\phi}_l}) Y(t). \tag{3.14}$$

Substituting it into (3.13), we further have the unified spatiotemporal model in the following way:

$$\begin{aligned} &(\boldsymbol{I} - \sum_{k=1}^{p}\sum_{l=0}^{\lambda_k} \widetilde{\boldsymbol{\phi}_{kl}} \boldsymbol{W}_l L^k)(1-L)^d (\boldsymbol{I} - \sum_{l=1}^{\lambda} \widehat{\boldsymbol{\phi}_l}) Y(t) \\ =&(\boldsymbol{I} - \sum_{k=1}^{q}\sum_{l=0}^{m_k} \widetilde{\boldsymbol{\theta}_{kl}} \boldsymbol{W}_l L^k)\varepsilon_t. \end{aligned} \tag{3.15}$$

For simplicity, we define

$$\begin{aligned} \boldsymbol{\phi}_{\pi,\tau,\lambda_1} &= \boldsymbol{I} - \sum_{l=1}^{\lambda} \widehat{\boldsymbol{\phi}_l}, \\ \boldsymbol{\phi}_{p,\lambda_2} &= \boldsymbol{I} - \sum_{k=1}^{p}\sum_{l=0}^{\lambda_k} \widetilde{\boldsymbol{\phi}_{kl}} \boldsymbol{W}_l L^k, \\ \nabla^d &= (1-L)^d, \\ \boldsymbol{\theta}_{q,m} &= \boldsymbol{I} - \sum_{k=1}^{q}\sum_{l=0}^{m_k} \widetilde{\boldsymbol{\theta}_{kl}} \boldsymbol{W}_l L^k. \end{aligned} \tag{3.16}$$

Finally, (3.15) can be rewritten as

$$\boldsymbol{\phi}_{\pi,\tau,\lambda_1}\boldsymbol{\phi}_{p,\lambda_2}\nabla^d Y(t) = \boldsymbol{\theta}_{q,m}\varepsilon_t \tag{3.17}$$

In our model, we put the physical factors potentially affecting such spatiotemporal correlation in the component $\boldsymbol{\phi}_{\pi,\tau,\lambda_1}$, which is independent of $\boldsymbol{\phi}_{p,\lambda_2}$ and $\boldsymbol{\theta}_{q,m}$. Besides, π reflects the trip distribution between adjacent links, and τ reflects the travel time delay between links in terms of the travel speed and route length; λ_1 reflects the number of spatially correlated links surrounding a link of interest. In this case, the accuracy of traffic flow prediction greatly relies on the estimation of $\boldsymbol{\phi}_{\pi,\tau,\lambda_1}$, relies to a lesser extent on $\boldsymbol{\phi}_{p,\lambda_2}$ and $\boldsymbol{\theta}_{q,m}$. The term "unified" in our proposed model is mainly manifested in the following aspects: 1) a day is divided into different time periods (e.g. peak and off peak hours) where traffic state in each time period can be regarded as static. The prediction model (3.17) in different time periods is identified by only adjusting $\boldsymbol{\phi}_{\pi,\tau,\lambda_1}$, which is estimated using the historical traffic data from the same time period of different days. 2) $\boldsymbol{\phi}_{p,\lambda_2}$ and $\boldsymbol{\theta}_{q,m}$ are required to be estimated once only based on traffic data and $\boldsymbol{\phi}_{\pi,\tau,\lambda_1}$ in any time period of the day. spatial autocorrelation function (SACF) and spatial partial ACF (SPACF) are applied to estimate $\boldsymbol{\phi}_{p,\lambda_2}$ and $\boldsymbol{\theta}_{q,m}$. Consequently, the challenging problem in model identification is the determination of λ_1, τ, and π, which will be further discussed in the next section.

3.2 Methodology for Parameter Estimation

In this section, we first propose the kernel strategies to estimate τ and π. After that, a BFS based algorithm is proposed to estimate λ_1 as well as τ and π. Finally, the computational complexity of model construction is analyzed.

3.2.1 Time-varying Lags τ

Consider two detector stations A and B with distance S where the vehicles keep a stable average speed v, then approximately $t = S/v$ is needed for vehicles to travel from B to A. In other words, the traffic flow collected at station A is strongly correlated with that at B t time ago. Thus the temporal lag with the maximum correlation should be $\tau = [t/t_{lag}]$ where t_{lag} is the length of one temporal lag. As v is time-varying, τ will change over the time. Therefore, we name τ as time-varying lag.

In (3.6), $\tau_{i,j}^{P_z}$ can be abbreviated as $\tau_{i,j}$ if the length of P_z is $l = 1$. $\tau_{i,j}$ is the time-varying lag between two adjacent links and estimated by

$$\tau_{i,j} = \frac{S_{i,j}}{v_{i,j} t_{lag}}. \tag{3.18}$$

where $S_{i,j}$ is the distance between V_i and V_j along the road, $v_{i,j}$ is the average traffic speed from link i to link j. Here the situation that $\tau_{i,j}$ may not be an integer is ignored for simplicity.

As a matter of fact, $v_{i,j}$ is the space mean speed (SMS). However, the speed collected by detectors is mostly the time mean speed (TMS) [2, 36]. To derive the SMS from the TMS, we use the method proposed by Jiang et al. [36]. The details are illustrated in Appendix B.

In the case that the length l of P_z is $l > 1$, the physical significance of $\pi_{i,j}^{P_z}$ within a sampling time interval t, denoted as $\tau_{i,j}^{P_z,t}$ can be interpreted as the sum of the delay caused by the travel time from link i to link j upon the path P_z. We use Ω to denote a set of time period clusters where the label of a cluster represents a specific time period of the day. We divide the successive time intervals of a day $T = \{1, 2, 3, ...\}$ into different clusters where the successive time intervals in a cluster compose a time period $T_m^n \in \Omega_n \subseteq \Omega$. As the classification algorithm is not the focus of this research, we use the ISODATA algorithm given in [24], or roughly make a

division according to the observation of the traffic flow data variation. After that, we have $\pi_{i,j}^{P_z}$ within a specific time period of day by

$$\tau_{i,j}^{P_z} = \lceil \frac{\sum_{t \in T_m^n} \tau_{i,j}^{P_z,t}}{|T_m^n|} \rceil, \tag{3.19}$$

where $\lceil x \rceil$ is the smallest integer that is greater than or equal to x. Further note that, the absence and breakdown of traffic detectors causes data missing, e.g., traffic speed and traffic flow. Thus, the aforementioned way to estimate $\tau_{i,j}^{P_z,t}$ is not available in this situation. Consider traffic flow or traffic speed data are not observable in a link $V_{miss} \in V$, we use a KNN based method [123] to estimate the TMS v_{miss} at V_{miss} by $v_{miss} = \sum_{k=1}^{K} v_k/K$, where $v_k, k \in K$ is the TMS at the k-th nearest links of V_{miss} ordered with respect to the Euclidean distance.

3.2.2 Turning Rate Estimation

In (3.2), $\pi_{i,j}$ is a special case of $\pi_{i,j}^{P_z}$ (in (3.6)) in the case that the length of P_z is $l = 1$. Indeed, the physical significance of $\pi_{i,j}^{P_z}$ is the ratio of the traffic attached to V_j with the traffic produced in V_i and traveling in P_z. Due to the fact that the turning rates at different intersections along a path are i.i.d., a simple way to estimate $\pi_{i,j}^{P_z}, l > 1$ is the accumulation of the turning rate between any two adjacent links in the path P_z from link i to link j. However, such estimation method is an intuitive, but not a general approach since a prior knowledge of the turning rate between any two adjacent links are needed. As the estimation of turning rate between two adjacent links V_i and V_j is closely correlated with the traffic at these two vertices, it is hard to infer $\pi_{i,j}$ once there is data missing in any link of V_i and V_j.

To overcome the aforementioned problem, we come up with a method motivated by the gravity model that is widely used for estimating the trip distribution between two zones. More precisely, the principle of gravity model states that the number of trips between two traffic zones is directly proportional to the number of trip attractions generated by the destination zone and inversely proportional to a function of

travel time between the two zones [29]. Based on the gravity model, we estimate $\pi_{i,j}^{P_z}$ by the following three-steps procedure which only requires the traffic at both ends of a path P_z, rather than the traffic from each link in the path.

- Divide a path P_z into a sequence of concatenate sub-path by $P_z = \cup_s P_{z_s}$ where the links without measured data are distributed into each sub-path P_{z_s}, whereas the links at the both ends of P_{z_s} have measured data;
- Apply a modified gravity model to calculate the turning rate upon sub-path P_{z_s};
- $\pi_{i,j}^{P_z} = \prod_s \pi_{i,j}^{P_{z_s}}$.

Suppose a vertex $V_i \in V$, as well as $\widehat{V_i^{l-}}$ where each $V_j \in \widehat{V_i^{l-}}$ has measured data and there is a path P_z with length l from V_i to V_j. We use $\widehat{P_{i,j}^{l}}$ to denote the collection of paths from V_i to $\forall V_j \in \widehat{V_i^{l-}}$. The gravity model based method is formulated as follows:

$$y_{i,j}^{P_z} = y_i \left[\frac{y_j \mathcal{C}_{i,j}^{P_z} \mathcal{B}_{i,j}^{P_z}}{\sum_{P_z \in \widehat{P_{i,j}^{l}}} y_j \mathcal{C}_{i,j}^{P_z} \mathcal{B}_{i,j}^{P_z}} \right]. \tag{3.20}$$

Particularly, when $l = 1$, the turning rate between two adjacent links is calculated. We use t_{P_z} to denote the travel time of vehicles traveling along the path P_z and is calculated by $t_{P_z} = \tau_{i,j}^{P_z} \times t_{lag}$ based on (3.18). Thus the inverse function of travel time t_{P_z}, $\mathcal{C}_{i,j}^{P_z}$ in (3.20), can be obtained from the calibration process [29]. $\mathcal{B}_{i,j}$ is socioeconomic adjustment factor for the interchange between vertices V_i and V_j, and in this paper, $\mathcal{B}_{i,j} = 1$. Within a time period $T_m^n \in \Omega_n \in \Omega$, y_i, y_j and $y_{i,j}^{P_k}$ are defined in the following way:

$$\begin{gathered} y_i = \textstyle\sum_{t \in T_m^n} y_i(t), y_j = \sum_{t \in T_m^n} y_j(t) \\ y_{i,j}^{P_z} = \textstyle\sum_{t \in T_m^n} y_{i,j}^{P_z}(t). \end{gathered} \tag{3.21}$$

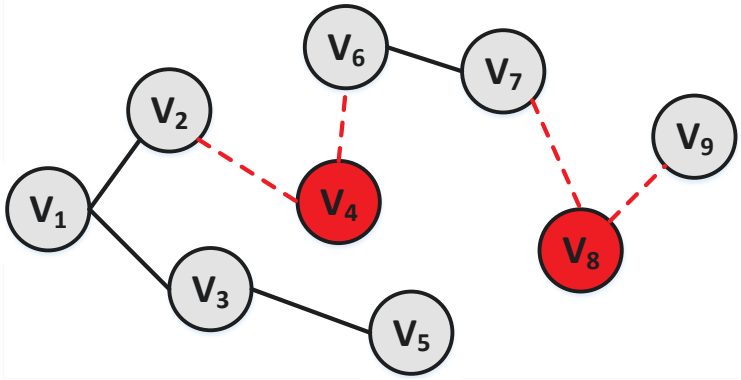


Figure 3.2 : An instance of turning rate estiamtion with incomplete data.

The objective of the gravity model is to estimate $y_{i,j}^{P_z}$ so that $\pi_{i,j}^{p_z}$ can be further estimated by $\pi_{i,j}^{p_z} = \frac{y_{i,j}^{P_z}}{y_i}$. We use an iterative procedure [29] to estimate y_j until convergence is reached:

$$y_{j,w} = \frac{y_j}{\sum_{i \in \mathcal{N}} \sum y_{i,j}^{P_z}} y_{j,w-1}. \tag{3.22}$$

In (3.22), w is the iteration number. Finally, we have $\pi_{i,j}^{p_z}$. To better understand the above estimation process, we give an example in Figure 3.2 where no detectors are configured at V_4 and V_8, causing the turning rates upon the dash and red lines can not be estimated directly. In this case, $P_z = P_{z_1} \cup P_{z_2}$ where P_{z_1} is the path from V_9 to V_6, whereas P_{z_2} is the path from V_6 to V_1. We first get the turning rate upon P_{z_1} is $\pi_{9,6}^{P_{z_1}}$, and we also get the the turning rate of P_{z_2} is $\pi_{6,1}^{P_{z_2}}$. Then we have $\pi_{9,1}^{P_z} = \pi_{9,6}^{P_{z_1}} \times \pi_{6,1}^{P_{z_2}}$.

3.2.3 Spatial Order λ_1 and Parameters Estimation Algorithm

To identify λ_1, as well as τ and π, a Breadth-First-Search (BFS) based algorithm is designed in Algorithm 3.1. In order to improve the efficiency, the estimation of τ, π and $\lambda_j, j \in \hat{\mathcal{N}}$ is executed in each vertex concurrently (line 1 to 21). With $\lambda_j, \forall j \in \hat{\mathcal{N}}$, λ_1 is calculated in a centralized way (line 22). With the determination

Algorithm 3.1: The estimation of τ, π, and λ_1

Input: τ, π **and** λ_j **for each link** $j \in \hat{\mathcal{N}}$

$P \leftarrow \varnothing$, $\lambda_j = 0$, $Q \leftarrow \varnothing$, $visited = 0$, $Q \leftarrow V_j$,$visited[j] = 1$

while $Q \neq \varnothing$ **do**

  $V_{temp} \leftarrow$the head in the Q

  V_i : there is an arc from V_i toV_{temp}

  while $V_i \neq \varnothing$ **do**

    if $visited[i] = 0$ **then**

      if *there is traffic data at* V_i **then**

        $\widehat{V_{temp}^{1+}} \leftarrow V_i$

        $P \leftarrow P_z$ from V_i to V_j with length l

        if $\lambda_j < l$ **then**

          $\lambda_j = l$

        end

        Estimate $\tau_{i,j}^{P_z}$ and $\pi_{i,j}^{P_z}$

      else

        $visited[i] = 1$

        $Q \leftarrow V_i$,$\widehat{V_{temp}^{1+}} \leftarrow V_i$

      end

      V_i : the next vertex that there is an arc from V_i to V_{temp}

    end

  end

end

Calculate λ_1 **using** (3.11)

of τ, π, and λ_1 , the uniform STARIMA model is set up according to the following three steps [69]:

- **Model Identification:** using STACF (space-time autocorrelation function) and STPACF (space-time partial autocorrelation function) to determine the maximum lags ($\{p, \lambda, q, m\}$) in the uniform STARIMA model.
- **Parameter Estimation:** estimating the model parameters ($\boldsymbol{\phi}_{p,\lambda_2}$ and $\boldsymbol{\theta}_{q,m}$) by non-linear optimization techniques;
- **Diagnostic Checking:** there are two phases in this process. In the first phase, the residuals will be examined in order to make the model adequately represents the data. In the second phase, it analyzes the statistical significance of the estimated parameters in order to avoid constructing a unduly complex (e.g., overfitting) model.

The parameters τ, π and λ_1 in the modified STARIMA model can be regarded as "hyper parameters" like those in deep learning model, e.g., the number of hidden layers. The difference is that these hyper parameters have physical meanings. Besides, with Algorihtm 1, these hyper parameters can be easily estimated (the algorithm complexity is analyzed in the following paragraph). Comparing with most studies where a lot of parameters have to estimate due to the fact that multiple models, particularly non-parametric models built in different time periods of the day, the distinguished advantage of our method is that we only need to adjust τ, π and λ_1 in different time periods of the day, rather than re-estimating all the parameters repeatedly. Thus, the accuracy complexity trade-off can be guaranteed.

We now analyze the computational complexity of parameters estimation in the STARIMA model. According to literature [35], Dave suggested that the computational complexity of identifying parameter p using ACF (autocorrelation function,

resp. parameter q using PACF, partial autocorrelation function) for the ARIMA model is $O(N_S N_l)$ where N_s is the number of samples from an observation and N_l is the number of time lags [35]. Unlike the ARIMA model, the parameters p and λ_ks in the STARIMA model are identified by STACF (resp. STPACF for q and m_ks). Thus, the computational complexity of calculating STACF (STPACF) between two links is $O((\mathcal{N}-1)N_l N_s)$ where $\mathcal{N}-1$ is the maximal spatial lag between two links. Consider any pair of links and the number of time periods n in a day, we have the computational complexity of parameters estimation in the STARIMA model is $O(nN_l N_s \mathcal{N}^3)$.

The computational complexity of parameters estimation for the unified spatiotemporal model mainly relies on the following two parts: 1) the identification of τ, π, and λ which relies on Algorithm 3.1 for paths searching, executed by the $\mathcal{N}$ vertices in parallel with computational complexity $O(\mathcal{N}^2)$. 2) the identification of parameters in STARIMA model that has the complexity is $O(N_l N_s \mathcal{N}^3)$. As a result, the total computational complexity is $O(n\mathcal{N}^2 + N_l N_s \mathcal{N}^3) = O((\frac{n}{\mathcal{N}} + N_l N_s)\mathcal{N}^2)$. Generally speaking, $\frac{n}{\mathcal{N}} + N_l N_s \ll nN_l N_s$ when a large amount of samples is considered at each observation.

3.3 Simulation and Discussion

3.3.1 Experimental Setup

In order to verify the performance of the proposed model, two datasets are used, thereafter referred as the dataset from one-dimensional freeway and the dataset from two-dimensional freeway incorporating on- and off-ramps (Fig.3.3 and 3.4)*. The reason for using different datasets is the need for exploring the impact of different

*The first set of data can be downloaded from: http://ngsim-community.org. The second set of data can be downloaded from: http://portal.its.pdx.edu.

road network topology on the prediction accuracy of the unified spatiotemporal model. For example, with the aid of the dataset from the two-dimensional network, we can clearly present the estimation of turning rate with the methods provided in Section 3.2.2. The dataset in the first group is sampled from six dual-loop detector

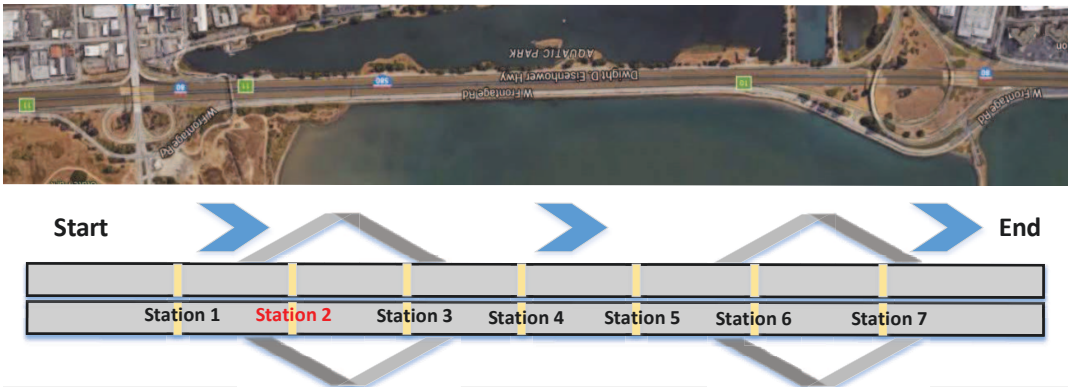


Figure 3.3 : The map and the topology of considered segment in I-80 freeway

stations deployed on a road segment of Interstate 80 (I-80) freeway in Emeryville, California, which are numbered by 1, 3, 4, 5, 6 and 7 (Fig.3.3). Furthermore, 10-days traffic data is recorded with sampling interval of 30 seconds ($t_{lag} = 30s$). We regard the mean traffic flow of every 3 data points as one data point. Thus, 960 (2880/3)/day $\times 9$ data points are available for training model, and the data in the last day are used for prediction.

The dataset in the second group is collected from I-205 NB Portland-area freeway. The freeway in Fig. 3.4 covers 10.09 miles (16.24km) including a major road with on- and off-ramps. In addition, the freeway is equipped with 14 detector stations to record the traffic traveling from north to south. Particularly, we select the data within 10 working days (Monday to Friday) from Sept. 19, 2011 to Sept. 30, 2011 with sampling interval of 20 seconds ($t_{lag} = 20s$). The locations of the detectors are marked by yellow (e.g., 1045) and orange lines (e.g., 5045) in the figure. The yellow lines are the detectors installed at the major road, while the orange lines are the

detectors installed at the entrance from the on-ramp to the major road. The station surrounded by the red circle means there is no available traffic data. We use the first 9-days data to train the model and the data in the last day to validate the prediction. Theoretically, there should be 4320 data at each station in one day. Unfortunately, there are some missing and dirty data inside. Hence, we use a commonly used way, named historical average, to replace the missing data by the average of the known values [73, 94].

We compare our proposed model (denoted as uSTARIMA) with other three approaches, respectively the STARIMA(p, q, λ, m) (denoted as STARIMA), multiple STARIMA based method (denoted as STARIMA*) in which the parameters and coefficients would be re-evaluated in different time periods of the day, and the BPNN method. The STARIMA and STARIMA* are both linear predictive method, while BPNN is a non-linear predictive method. We use a $4 \times 20 \times 1$ BPNN model including a hidden layer and an output layer to predict the traffic flow at each measurement point. There are 4 input nodes which respectively denote the traffic flow data collected from the same measurement at t, $t-10min$, $t-20min$ and $t-30min$. There are 20 nodes in the hidden layer and one node in the output layer. The initial weights are randomly distributed inside a range $[-0.12, 0.12]$ and the thresholds have initial values of 0. We use the sigmoid function as the active function. Besides, we set the momentum coefficient to be 0.7, and the learning rate to be 0.3. A gradient descent optimization algorithm is used to adjust the weights and thresholds by calculating the gradient of the loss function iteratively until the sum of squared errors is no more than the learning error set by 0.01. We use R language running on 64-bit Windows system with 4 CPUs and 16G RAM. With the aid of starma [†] and neuralnet packages [‡], we develop our uniform model as well as the other counterparts. Particularly,

[†]https://cran.r-project.org/web/packages/starma/starma.pdf

[‡]https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf

starma packages integrated three-stage iterative modeling procedure. In order to verify the prediction accuracy and the efficiency of the proposed scheme, the metrics of the mean square error (MSE), the mean absolute percentage error (MAPE) and running time are considered. More precisely, let $\hat{y}$ be the estimate of N-dimensional vector y, then MSE can be expressed as $MSE(\hat{y}, y) = 1/N \sum_{n=1}^{N} (\hat{y}_n - y_n)^2$, and MAPE is calculated by $MAPE(\hat{y}, y) = 1/N \sum_{n=1}^{N} |\frac{\hat{y}_n - y_n}{y_n}|$.

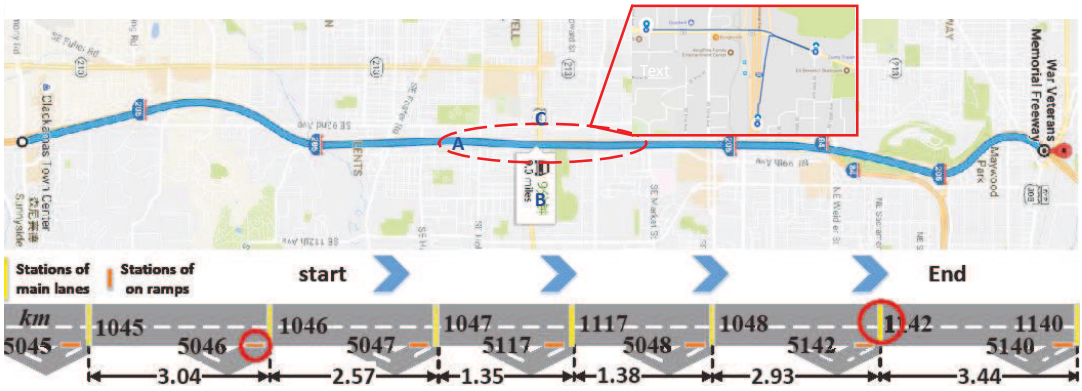


Figure 3.4 : The map and topology of I-205 NB freeway

3.3.2 Experimental Results for One-dimensional Freeway

According to the traffic data collected at stations 3 and 6 on I-80 freeway, we intuitively set $\Omega_1 \in \Omega$ (peak hour) by $\Omega_1 = \{T_1^1\}$ where T_1^1 covers the time period from 6:30am to 9:00am. Correspondingly, Ω_2 (off-peak hour) is the set of time periods outside the range of 6:30am-9:00am. Hereafter, we provide the MAPE/MSE of the traffic prediction at different time of the day in Table 3.2.

From the experimental results, we can observe that the best performance is the one achieved by STARIMA*. Such phenomenon can be explained by the fact that the simple road topology structure enables the time-varying spatiotemporal correlation can be successfully captured by re-estimating all the parameters ($\{p, q, \lambda, m\}$ and $\{\phi_{p,\lambda}, \theta_{q,m}\}$) of STARIMA* in each time period of the day. In comparison,

Table 3.1 : The time-varying lags between stations with spatial order $l \geq 2$

From	s_3	s_3	s_4
To	s_5	s_6	s_6
0:00am-6:00am	1	2	1
6:00am-9:00am	2	3	2
9:00am-16:00pm	1	2	1
16:00pm-18:00pm	2	3	2
18:00pm-24:00pm	1	2	1

Table 3.2 : The MAPE/MSE of one-day traffic flow prediction using uSTARIMA, STARIMA*, STARIMA and BPNN

St.	uSTARIMA	STARIMA*	STARIMA	BPNN
s_3	17.80%/206.33	14.86%/164.21	22.19%/245.37	31.52%/315.22
s_4	17.12%/191.25	15.84%/179.06	25.68%/259.24	24.75%/238.97
s_5	15.13%/178.59	14.92%/159.57	20.41%/209.73	30.21%/334.61
s_6	14.41%/136.27	12.65%/112.44	23.77%/142.57	22.45%/215.72

our proposed uSTARIMA model has a slight increase in prediction error ($\sim$ 3% of the measured value). The loss of accuracy is caused by that a nearly monotonous structure of uSTARIMA is set up in different time periods. In other words, there is no obvious difference between the parameters $\{\pi, \tau, \lambda_1\}$ of uSTARIMA in different time periods. For instance, the turning rate between any two adjacent links is a constant value "1" since there is no intersection in the study site. Further, as the road segment (between s_3 and s_6) is not long, we can find the maximal time-varying lag is 3 between s_3 and s_6 in peak hour from Table 3.1. On the contrary, the minimal time-varying lag is 1 between s_3 and s_5 in off-peak hour. The time-varying lags

between any adjacent stations are not presented in Table 3.1 since the values are smaller than 1, but approximately equal to 1 according to formulation (3.19). Also, $\lambda_1 = 3$ based on the graph model of the study site. The slight change of $\{\pi, \tau, \lambda_1\}$ has no significant impact on the estimation of $\phi_{\pi,\tau,\lambda_1}$. Thus, the performance of uSTARIMA is a little worse than STARIMA*. In practice, the forecasting accuracy of uSTARIMA is sensitive to the fluctuation of above three parameters $\{\pi, \tau, \lambda_1\}$. This can be observed from the experimental results on the basis of the study site in the second group, which we will illustrate in the next sub-section. In addition, comparing with the STARIMA and BPNN technique, unsurprisingly, the proposed technique achieves much better prediction accuracy. Particularly, the MAPE of the proposed technique is at least 5%, at most 15% better than that achieved by these two techniques. Fig. 3.5 shows the running time of each approach. From the fig-

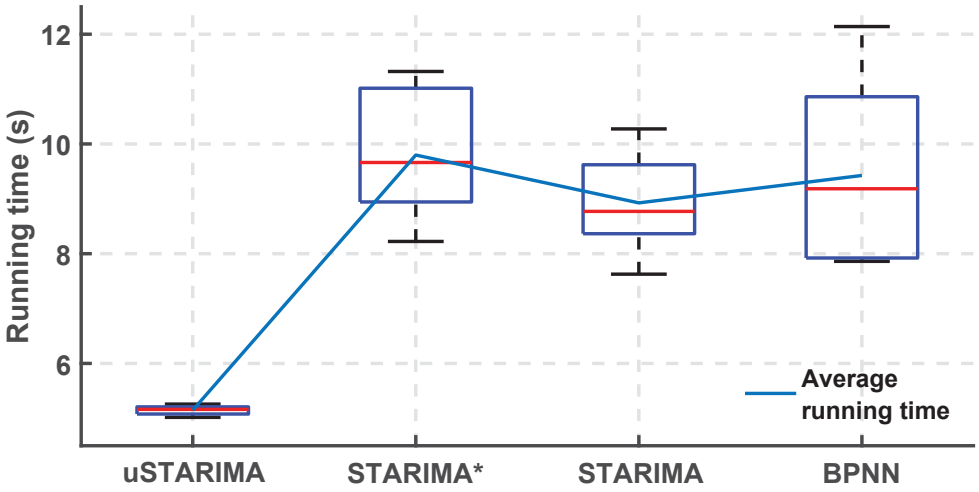


Figure 3.5 : The running time of STARIMA(Ξ), STARIMA*, ARIMA* and BPNN

ure, we know that the running time of uSTARIMA is much less than the other two methods, attributable to the unified model employed for traffic prediction during different time periods.

3.3.3 Experimental Results for Two-dimensional Network

Based on the traffic data collected at 6 stations on the major road of I-205 NB freeway, we intuitively divide a day into three time periods. Specially, $\Omega_1 = \{T_1^1, T_2^1\}$ where T_1^1 covers the time period from 6:00am to 9:00am and T_2^1 covers the time period from 16:00pm to 18:00pm. Correspondingly, Ω_2 consists of the set of time periods outside of T_1^1 and T_2^1 . Since the major road in the freeway is long, we divide it into a set of links where each detector station is distributed in one link. In this paper, we mainly provide the traffic prediction at each detector station on the major road.

We list the time-varying lags between two neighboring links in the major road in Table 3.3. It further verifies that the time-varying lag has a close relation with the distance of two stations, as well as different travel speeds during different time periods of the day. Then, we calculate the time-varying lags between stations by means of Algorithm 3.2.3. For instance, the time-varying lag between station 1045 and 1117 is 14 $(6+5+3)$ within 6:00am-9:00am, while 11 $(5+4+2)$ within 9:00am-16:00pm. As the vehicles coming from on-ramps will move into the major road, the turning

Table 3.3 : The time varying lag between two neighboring links on the major road in different time periods of the day

From	1045	1046	1047	1117	1048
To	1046	1047	1117	1048	1140
0:00am-6:00am	5	4	2	2	10
6:00am-9:00am	6	5	3	3	12
9:00am-16:00pm	5	4	2	2	10
16:00pm-18:00pm	6	5	3	3	13
18:00pm-24:00pm	5	4	2	2	10

Table 3.4 : The estimation of turning rates at the intersections of major road and off-ramps (gravity based methods/data-driven method)

From	1045		1046		1047		1117		1048	
To	1046	Off	1047	Off	1117	Off	1048	Off	1140	Off
0:00am-6:00am	0.72/-	0.28/-	0.81/0.78	0.19/0.22	0.77/0.69	0.23/0.31	0.86/0.80	0.14/0.20	0.87/0.86	0.13/0.14
6:00am-9:00am	0.73/-	0.27/-	0.83/0.79	0.17/0.21	**0.48/0.54**	**0.52/0.46**	0.86/0.79	0.14/0.21	0.89/0.90	0.11/0.10
9:00am-16:00pm	0.71/-	0.29/-	0.88/0.90	0.12/0.10	**0.56/0.55**	**0.44/0.45**	0.88/0.84	0.12/0.16	0.88/0.79	0.12/0.21
16:00pm-18:00pm	0.73/-	0.27/-	0.89/0.76	0.11/0.24	**0.43/0.47**	**0.57/0.53**	0.79/0.77	0.21/0.23	0.84/0.73	0.16/0.27
18:00pm-24:00pm	0.71/-	0.29/-	0.89/0.88	0.11/0.12	0.79/0.66	0.21/0.34	0.88/0.85	0.12/0.15	0.91/0.86	0.09/0.14

Table 3.5 : The MAPE/MSE of one-day traffic flow prediction using uSTARIMA, STARIMA*, STARIMA and BPNN

St.	uSTARIMA	STARIMA*	STARIMA	BPNN
1045	24.21%/165.57	17.08%/159.75	25.14%/202.69	41.14%/452.84
1046	19.29%/184.91	18.32%/175.76	23.72%/195.07	28.43%/394.25
1047	**12.57%/103.54**	12.78%/119.49	22.60%/154.25	34.26%/405.67
1117	35.95%/413.51	29.97%/297.06	46.81%/594.62	45.28%/481.24
1048	**15.72%/116.64**	16.54%/139.72	17.19%/130.84	35.11%/375.22
1140	19.03%/121.13	15.46%/108.54	24.11%/115.07	37.37%/326.33

rate at the intersection between on-ramps and major road is equal to 1. However, the vehicles at the intersection of off-ramps and major road have two alternatives. One is leaving the freeway through off ramps, and the other one is to keep traveling straightly on the major road. The results of turning rate estimation are presented in Table 3.4. Table 3.4 presents the turning rates respectively estimated by gravity based method and data-driven method. We regard the data-driven based method as the "actual scenario" where the turning rate at an intersection is calculated by the ratio of "the traffic flow streaming into the off-ramps" to "traffic flow traveling from the major road". As there is no detector configured in the off-ramps, we cannot obtain the traffic flow streaming into the off-ramps directly. However, we can roughly estimate it using the traffic flow data collected at two adjacent detector stations as well as the stations configured in the on-ramp between these two stations. For instance, suppose we have time-spaced traffic flow data at station 1046, 1047 and 5047, respectively y_{1046}, y_{1047} and y_{5047}. Then the traffic flow streaming into the off-ramps between station 1046 and 1047 is calculated by "$y_{off} = y_{1046} - (y_{1047} - y_{5047})$" . Given a time period T, we estimate turning rate by "$\frac{\sum_T y_{off}}{\sum_T y_{1046}}$". Note, there is no

data at station 5046, thus, we can only use the data-driven method to estimate the turning rates at the other intersections. To save space, the values of the turning rates estimated by both methods are rounded off to the two decimal places. From Table 3.4, we can observe that the results obtained from gravity based methods are approximately the same as the ones calculated by data-driven method. As we have mentioned in Section IV-B, one advantage of gravity based model is that it can be used to estimate the turning rate even there is missing data in some roads such as the estimation turning rates between station 1045 and off-ramp at the second column (with '-'). Except for the turning rates labeled in red, all the other turning rates show that above 70% vehicles will keep traveling on the major road. As for the turning rates between station 1047 and the downstream off-ramp, we observe that the off-ramp is connected with a road named "SE Powell Blvd" across the segment between station 1047 and 1117 (circled by the dashed line in red). In [93], Stoll et al. indicated that Powell Blvd road was a major arterial road in the Portland metropolitan area and carried between 45,000 and 30,000 vehicles a day. The large traffic volume in Powell Blvd road implies that a lot of vehicles will leave the major road and move into Powell Blvd road (e.g. the vehicles traveling from A to B or from A to C). Therefore, in each time period of the daytime (from 6:00 am to 18:00 pm), the turning rate between station 1047 and 1117 are less than the ones estimated between any other pairs of stations. The discrepancies in the estimated turning rates further verify our idea that road trip distribution has a critical influence on the analysis of spatiotemporal correlation. With turning rate and time-varying lag estimated above, we predict the traffic flow at 6 stations (without station 1142). From Table 3.5, we can see that the MAPE of our proposed model is at most $\sim 6\%$ (at stations 1046 and 1117) lower than STARIMA*. Note that the forecasting results obtained from our proposed model have the best accuracy. This can be illustrated that the time-varying spatiotemporal correlation affected by the frequent variation

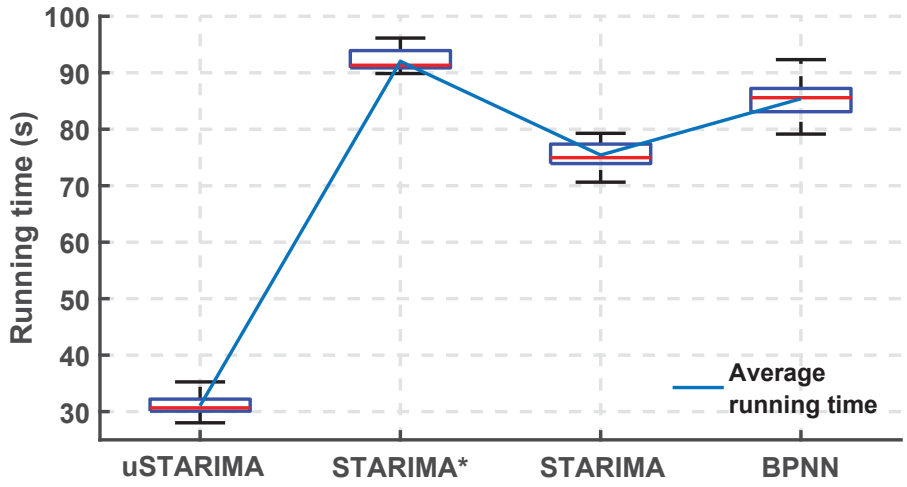


Figure 3.6 : The running time of uSTARIMA, STARIMA*, STARIMA and BPNN

of travel speed and trip distribution in the study site of I-205NB freeway can be better captured by the introduced parameters $\{\pi, \tau, \lambda_1\}$ in our method. For instance, given two stations 1047 and 1048, the gap between the time-varying lag in peak and off-peak hours can be 7 (30 in peak hour from 16:00 pm to 18:00 pm and 23 in off-peak hour from 9:00 am to 16:00 pm). Distinctly different from uSTARIMA, the determination of lags in STRAIMA* is by means of STACF and STPACF which depend on the assumption that we are comfortable making with respect to the constancy of the trend in the data. It is difficult to select accurate number of lags. Thus, the forecasting accuracy of STARIMA* will be reduced in some cases, e.g., station 1047 and 1048. Unsurprisingly, in the worst case there is $\sim 22\%$ (at station 1117) gap between the MAPE of uSTARIMA and BPNN, and at worst $\sim 10\%$ (at station 1047) gap between the MAPE of uSTARIMA and STARIMA.

In Fig. 3.6, we present the running time of different methods. It is clear to see that less time is consumed for our proposed model, which is consistent with the result in Fig.3.5. Based on the results in Table 3.5 and Fig.3.6, it is sufficient to say that our proposed model is also available for the two-dimensional road network.

3.4 Summary

In this chapter, we developed a unified spatiotemporal model, which does not need a complete re-design and calibration of the prediction model for short-term traffic flow prediction during the day. In the model, the spatiotemporal traffic correlation is captured by the turning rate at the intersections, as well as the time-varying lag which is formulated as a function of the spatial separation and the travel speed between two measurement points. Fundamentally, a better performance is achieved because, instead of using a black-box approach to model the traffic correlations, the proposed method explicitly takes into account the road topology, trip distribution and travel speed and offers a physically intuitive approach to capturing the spatiotemporal correlation between traffic at different locations. In this sense, a deeper insight revealed through our work is that by incorporating the knowledge of the underlying road topology into traffic prediction, a better accuracy can be achieved.

Chapter 4

Estimation of Link Travel Time Distribution With Limited Traffic Detectors

In this chapter, we present a novel methodology to estimate link travel time distributions (TTDs) using end-to-end (E2E) measurements detected by the limited traffic detectors at or near the road intersections. As it is not necessary to monitor the traffic in each link, the proposed estimator can be readily implemented in the real life. The technical contributions of this paper are as follows: First, we employ the kernel density estimator (KDE) to model link travel times instead of parametric models, e.g., Gaussian distribution. It is able to capture the dynamic of link travel times that vary with the change of road conditions. The model parameters are estimated with the proposed C-shortest path algorithm, K-means based algorithm, as well as expectation maximization (EM) algorithm. Second, to reduce the complexity of parameter estimation, we further propose a Q-opt and an X-means based algorithm. Lastly, we validate our proposed method using a dataset consisting of 3.0e+07 GPS trajectories collected by the taxicabs in Xi'an, China. With the metrics of Kullback Leibler and Kolmogorov-Smirnov test, the experimental results show that the link TTDs obtained from our proposed model are in excellent agreement with the empirical distributions, provided that $\sim 70\%$ of the intersections are equipped with traffic detectors.

The organization of the paper is as follows: We introduce the proposed model in Section 4.1, followed by the methods of parameter estimation in Section 4.2. We explore the performance of the proposed model in Section 4.3. Lastly, we draw the summary in Section 4.4.

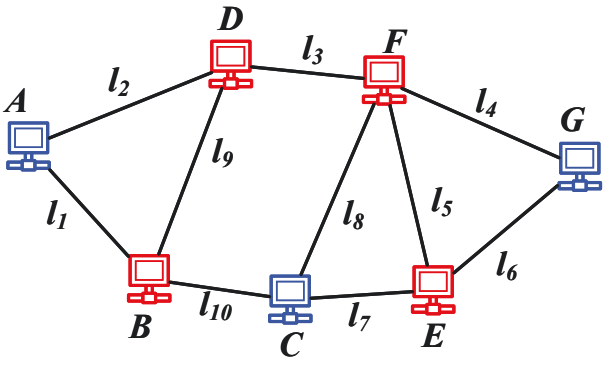
4.1 System Model

4.1.1 Network Tomography

To illustrate the principle of network tomography, a concrete example is given in Fig.4.1 ,where there are 10 links, denoted by $\{l_i|i = 1, 2, ..., 10\}$. The E2E measurements are taken by the beacons configured at $\{A, C, G\}$. Suppose the packets are transmitted through the routes $r_j \subseteq \{r_1, r_2, r_3, r_4, r_5\}$ (on the upper right of Fig. 4.1), the E2E measurements on each route are denoted by $Y = \{y_1, y_2, y_3, y_4, y_5\}$. $\forall y_j \in Y$ can be formulated as $y_j = \sum_{i=1}^{10} w_{ji}x_i$, where x_i is the delay on l_i; $w_{ji} = 1$, if l_i is covered by r_j, otherwise $w_{ji} = 0$. Given a route matrix W where each row represents a route and each column represents a link (on the bottom rig t of Fig. 4.1), we formulate the delays on all the routes as:

$$Y^{\mathrm{T}} = WX^{\mathrm{T}}, \tag{4.1}$$

where $X = \{x_i|i = 1, 2, ..., 10\}$, Y^{T} and X^{T} are the transposes of Y and X respectively. A large number of methods has been proposed to get the solution of X [26, 114].



End to end measurements

r_1: A–> D–> F–> G
r_2: A–> D–> B–> C
r_3: A–> B–> D–> F–> G
r_4: C–> F–> E–> G
r_5: C–> E–> F–> G

***Route Matrix*: W**

Links	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
r_1	0	1	1	1	0	0	0	0	0	0
r_2	0	1	0	0	0	0	0	0	1	1
r_3	1	0	1	1	0	0	0	0	1	0
r_4	0	0	0	0	1	1	0	1	0	0
r_5	0	0	0	1	1	0	1	0	0	0

Figure 4.1 : An instance of network tomography

4.1.2 Kernel Density Estimator

The probability density function (PDF) of the kernel density estimator (KDE) is defined as:

$$p(\boldsymbol{x}) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x_i - \boldsymbol{x}}{h}), \tag{4.2}$$

where $\boldsymbol{x}$ is the random variable, n is the number of samples, $h > 0$ is called the smoothing bandwidth that controls the amount of smoothing, x_i is the i-th sample, $K(\boldsymbol{x})$ is named the kernel (function) that is generally a smooth and symmetric function. There are various choices among kernels, such as uniform, triangle, Gaussian, and Epanechnikov kernels. The best fitting performance (lowest mean square error) is obtained with the Epanechnikov kernel. However, it will reduce the estimation efficiency. The fitting performances of uniform, triangle, and Gaussian kernels are similar. Therefore, for the sake of mathematical analysis, in this paper, we use the Gaussian kernel [85]. In particular, $K(\frac{x_i - \boldsymbol{x}}{h})$ follows the standard normal distribution of $\mathcal{N}(0, 1)$. Equivalently, we can rewrite (4.2) as follows:

$$p(\boldsymbol{x}) = \frac{1}{nh}\sum_{i=1}^{n} \mathcal{N}(\boldsymbol{x}|u_i, \sigma_i^2), u_i = x_i, \sigma_i = h. \tag{4.3}$$

4.1.3 KDE Based Model

To help readers keep track of symbols' meanings, we clarify the major notations in Table 1. Moreover, we provide a flowchart in Fig.4.2 to describe our proposed method. It includes:

- *Model building*: Given a study site, we use KDE to model travel time distributions across all the links (Section 4.1.2).
- *Parameter estimation*: The number of model parameters is closely related to the placement of traffic detectors. To this end, we design a C-shortest path algorithm (Section 4.2.1) with which the maximal number of paths between

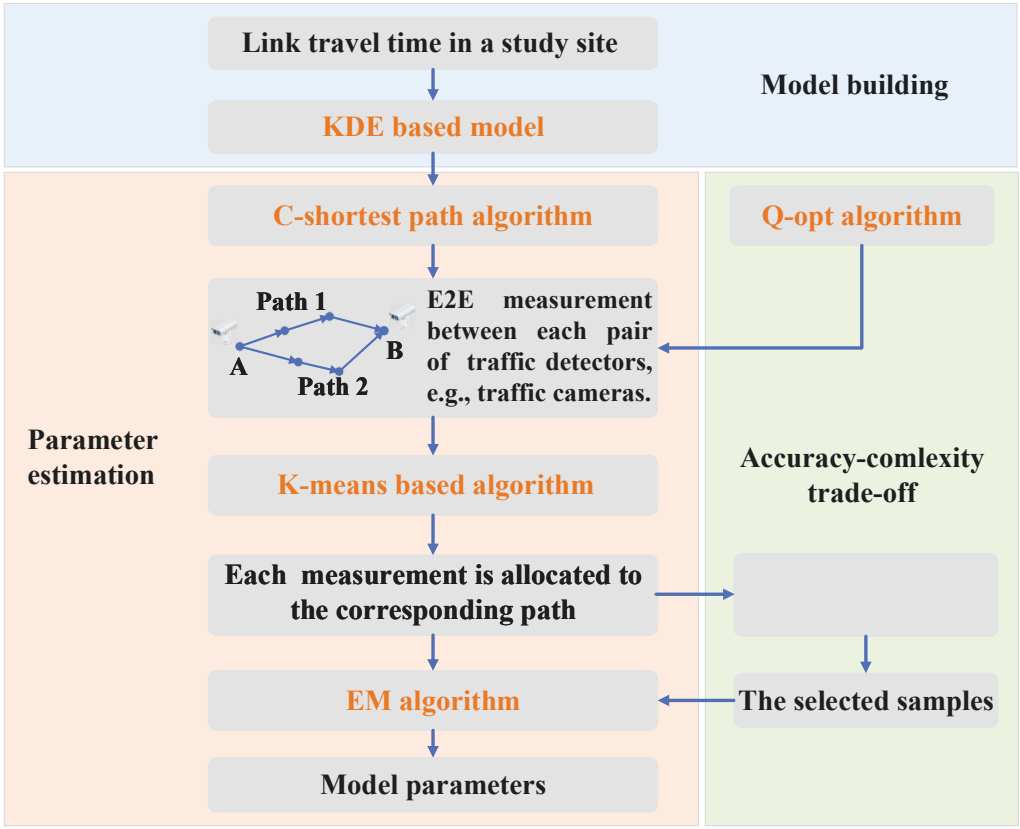


Figure 4.2 : Flowchart of the proposed method

any pair of traffic detectors is C. As there may be $C > 1$ paths between two traffic detectors, it is not clear which path an E2E measurement is collected from. Therefore, we propose a data allocation strategy based on K-means algorithm (Section 4.2.2). Lastly, EM algorithm (Section 4.2.3) is implemented to estimate the parameters.

- *Accuracy-complexity trade-off*: The complexity of EM algorithm depends on the number of links and the number of E2E measurements between two traffic detectors (the details will be illustrated in Section 4.2.4). To guarantee the accuracy-complexity trade-off, we first design a Q-opt algorithm so that the maximal number of links in a path is Q. It is executed together with the C-shortest path algorithm. To filter the the E2E measurements that contribute

little to the parameter estimation, we propose an X-means based sampling algorithm, executed after K-means based algorithm.

Table 4.1 : Symbols Table

Symbols	Significance
$G = \{V, E\}$	The digraph model of the road network
V_{meas}	The set of vertices deployed with traffic detectors (measurement points)
V_{unmeas}	The set of vertices without traffic detectors
r	A path between a pair of measurement points
d_r	The number of edges (links) in r
$\boldsymbol{t}_{e_k}$	The random variable of travel time for link e_k
$\boldsymbol{t}_r$	The random variable of travel time for the path r
t (t_r)	An E2E measurement (collected from the path r)
M	The number of pairs of measurement points
R_m	The set of the paths between the m-th pair of measurement points
R	$\{R_m \vert \forall m \in M\}$
$\mathbb{R}$	$\{r_j \vert r_j \subseteq R_m, \forall m \in M\}$
T_m	The set of samples collected by the m-th pair of measurement points during a time interval
T	$\{t \in T_m \vert \forall m \in M\}$
$\hat{\mathrm{T}}$	The subset of T after sampling with X-means based algorithm
$p_{t\vert r_j}$	A binary variable $\{0, 1\}$ denotes whether t is collected on r_j ($p_{t\vert r_j} = 1$) or not ($p_{t\vert r_j} = 0$)
$P_{t\vert R_m}$	$\{p_{t\vert r_j} \vert \forall r_j \subseteq R_m, m \in M\}$
$\mathbf{P}_{\mathrm{T\vert R}}$	$\cup P_{T_m \vert R_m}, \forall m \in M$
$\mathbf{P}_{\mathrm{T\vert \mathbb{R}}}$	$\{p_{t\vert r_j} \vert t \in \mathbf{T}, r_j \subseteq \mathbb{R}\}$

Continued on next page

Table 4.1 continued from previous page

Symbols	Significance
$\mathbb{T}$	$\{t \mid t \in \mathbf{T}, p_{t\mid r_j} = 1\}$
n_{e_k}	The number of vehicles traveling in e_k during a time interval
h_{e_k}	The bandwidth of the KDE model for e_k
$\mu_{e_k,i}$	The travel time for the i-th vehicle traveling in e_k
μ_{e_k}	$\{\mu_{e_k,i} \mid i \in n_{e_k}\}$
Θ_{e_k}	$\{n_{e_k}, h_{e_k}, \mu_{e_k}\}$
Θ_{r_j}	$\{\Theta_{e_k} \mid e_k \in r_j\}$
$\mathbf{\Theta}_{\mathrm{R}}/\mathbf{\Theta}_{\mathbb{R}}$	$\{\Theta_{e_k} \mid e_k \in E\}$
W	The route matrix where each row represents a path and each column represents an edge.
B_W	A basis of W
$\mathbf{B}_W$	The set of bases of W
$\boldsymbol{y}_{r_j}$	The latent variables where $\forall y_z \in \boldsymbol{y}_{r_j}$ satisfies $y_z = \{0, 1\}$ and $\sum_{y_z \in \boldsymbol{y}_{r_j}} y_z = 1$
$\gamma_{r_j}(y_z)$	The probability $p(y_z = 1 \mid t_{r_j})$

To illustrate the advantage of KDE model, we analyze the cumulative density functions (CDFs) of travel times on a randomly selected link in Xi'an road network. The details of data will be illustrated in Section 4.3. In the analysis, we divide a day into 48 time intervals, each of which is half an hour (e.g., time interval 1 represents time period from 00:00:00am to 00:30:00am). We consider two road conditions: free flow and congestion. To identify the road condition in each time interval, we use the method proposed by Nguyen et al. [66]. specifically, a road is considered to be congested, if the mean travel time in a time period is greater than n-th percentile of the mean travel times in the whole time intervals. In this paper, we use $n = 80$. The red line in Fig.4.3 is the 80th percentile of all the mean travel times in 48 time

intervals. The congestion happened in the time intervals where the points are above the red line. Otherwise, the road is in the state of free flow.

In Fig. 4.4a and Fig. 4.4b, we present the CDFs of the empirical data, Opt-KDE, Gaussian distribution and log-normal distribution under the road conditions of congestion and free flow respectively. Opt-KDE represents the KDE with optimal bandwidth estimated by the biased cross-validation method proposed by Scott and George [81]. From Fig. 4.4, we can observe that the CDF of Opt-KDE matches the empirical data better than the other two models. Furthermore, we use the KS (Kolmogorov-Smirnov) test to measure the similarity between the CDFs of two distributions. The null hypothesis of the KS test is that the two distributions are the same. Given a significance level ($\alpha = 0.01$), we reject the null hypothesis, if the maximal distance between two CDFs is greater than the critical value, $c(\alpha)$, defined by:

$$c(\alpha) = \sqrt{-\frac{1}{2}\ln\alpha} \cdot \sqrt{\frac{n+m}{nm}}, \tag{4.4}$$

where n is the size of empirical data and m is the size of data used for probabilistic model estimation. Moreover, we use $D_{em,es}$ to denote the maximal distance between the CDFs of empirical data and estimated distribution. From Table 4.2, we find that the Gaussian model is rejected in the case of free flow and both Gaussian and log-normal models are rejected in the case of congestion. Moreover, with the bandwidth $2s \leq h \leq 5s$, KDE model also fit the data better than the Gaussian and log-normal models. Similar results can also be obtained based on the data in other links.

We model a road network as a digraph. Specifically, we partition the road network into a set of *links* where each link is an one-way road segment bounded by two road intersections and there is no intersection within a link. Drawing from the graph theory, the digraph model is represented as $G = (V, E)$ where V is the set of vertices and E is the set of directed edges. Each vertex $V_i \in V$ represents an intersection. There exists an edge $e_{ij} \in E, e_{ij} = (V_i, V_j)$ if there is a link with

Table 4.2 : The KS test based on different probabilistic models with significance level $\alpha = 0.01$.

$D_{em,es}$	Opt-KDE	KDE ($h = 3s$)	KDE ($h = 4s$)	KDE ($h = 5s$)	Gaussian	log-normal
Free flow ($c(\alpha) = 0.100$)	0.031	0.039	0.053	0.072	0.167 (reject)	0.100
Congestion ($c(\alpha) = 0.088$)	0.022	0.030	0.029	0.022	0.259 (reject)	0.169 (reject)

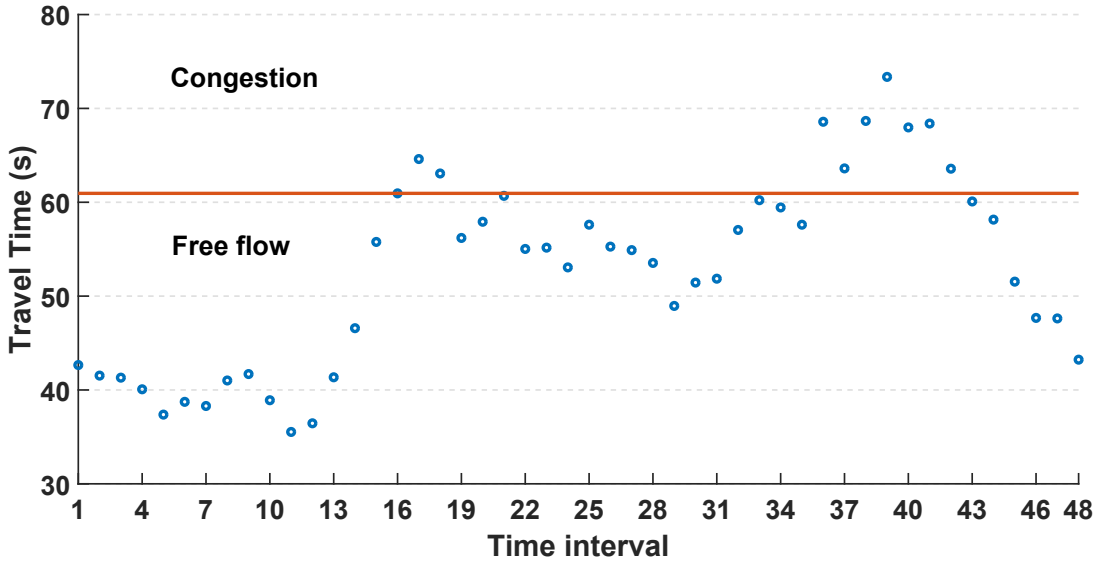
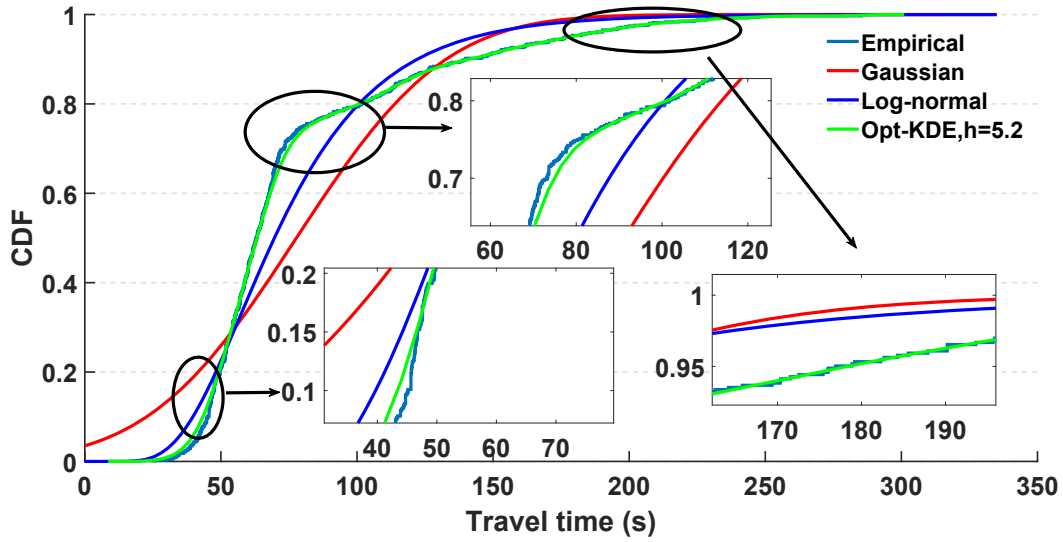


Figure 4.3 : The average travel time on a road in each time interval. The red line is the 80% of all the points.

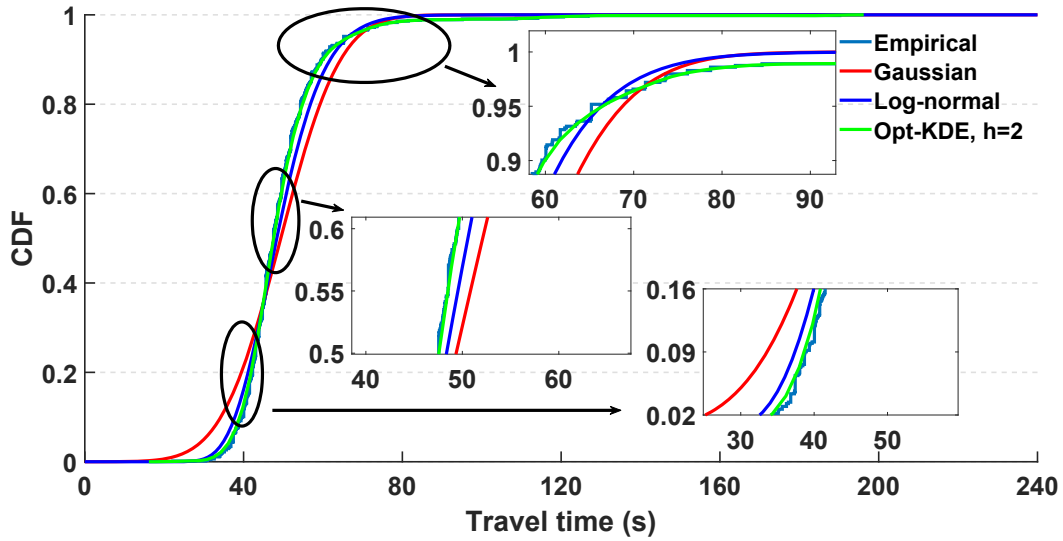
traveling direction from V_i to V_j. We name a vertex V_i as a measurement point if there are observations detected by the traffic detectors like traffic cameras at V_i. Then $V = \{V_{meas}, V_{unmeas}\}$ where V_{meas} is the set of measurement points and $V_{unmeas} = V - V_{meas}$. Obviously, the TTD of e_{ij} can be estimated if both end points of e_{ij}, $V_i, V_j \in V_{meas}$. However, in the real life, it is impractical to cover $\forall V_i \in V$ with traffic detectors. As a result, there is always a sequence of links between two measurement points. With the principle of graph theory, we define a travel route between one intersection and another as a path, denoted by:

$$r = \{e_1, e_2, ..., e_{d_r}\}, \tag{4.5}$$

where d_r is the number of links and the edges in r are all distinct from each other. Given a path r between $V_i, V_j \in V_{meas}$, we can obtain the travel time on r with the observations at V_i and V_j. For instance, a vehicle travels from V_i to V_j through r and is captured by the traffic cameras at V_i and V_j at time t_1 and t_2, then the travel time $t = t_2 - t_1$. In this paper, we also name t as an E2E measurement.



(a) CDFs under congestion (8:30am-9:00am, the 18th time interval in Fig.4.3)



(b) CDFs under free flow (14:00pm-14:30pm, the 29th time interval in Fig.4.3)

Figure 4.4 : CDFs based on empirical data, Opt-KDE, Gaussian and log-normal models under the congestion and free flow respectively

Consider the situation that the positions of the traffic data collected by some traffic detectors are not exactly located at the road intersections, but somewhere nearby, e.g., GPS data collected by the probe vehicles. We use a distance and time proportion method to estimate E2E measurements. More details will be illustrated in Section 4.3. We use the bold-faced letter $\boldsymbol{t}$ to represent a random variable of travel time. The objective of our work is to estimate the distribution of $\boldsymbol{t}_{e_k}$ for $\forall e_k \in E$ with the E2E measurements detected by the limited traffic detectors.

We assume the travel times for the vehicles traveling in different links are spatially independent. Meanwhile, we assume different vehicles traveling in the same link will experience independent travel times. We respectively term these two assumptions as spatial independence and temporal independence. In practice, travel times in the links are generally spatially and temporally correlated to a greater or lesser extent. However, these correlations are usually not strong enough. In addition, ignoring dependencies can also have benefits on the analysis. For instance, in [105], to simply the objective function of path travel time estimation, Wang et al. assumed the travel times on different links are independent. Based on the above analysis, it is sufficient for us to use these assumptions to derive the estimates of TTD.

We model the distribution of $\boldsymbol{t}_{e_k}$ with KDE as follows:

$$p(\boldsymbol{t}_{e_k}|\Theta_{e_k}) = \frac{1}{n_{e_k} h_{e_k}} \sum_{i=1}^{n_{e_k}} \mathcal{N}(\boldsymbol{t}_{e_k}|\mu_{e_k,i}, h_{e_k}^2), \tag{4.6}$$

where $\Theta_{e_k} = \{n_{e_k}, h_{e_k}, \mu_{e_k}\}$ is the set of parameters. More precisely, n_{e_k} is the number of vehicles traveling through e_k during a time interval, h_{e_k} is the bandwidth and $\mu_{e_k} = \{u_{e_k,i}|i = 1, 2, ..., n_{e_k}\}$ where $u_{e_k,i} \in \mu_{e_k}$ is the travel time when the i-th vehicle traverses e_k. As $\boldsymbol{t}_r = \sum_{k=1}^{d_r} \boldsymbol{t}_{e_k}$, the distribution of $\boldsymbol{t}_r$ conditioned on Θ_{e_k} can be parameterized as follows:

$$p(\boldsymbol{t}_r|\Theta_r) = p(\boldsymbol{t}_{e_1}|\Theta_{e_1}) * ... * p(\boldsymbol{t}_{e_{d_r}}|\Theta_{e_{d_r}}), \tag{4.7}$$

where $*$ represents the convolution operation and $\Theta_r = \{\Theta_{e_k} | k \in d_r\}$.

In the network tomography, the transmission route of a packet is always known. However, in our work, the path r where an E2E measurement is collected is usually unknown because of the following two reasons: i) the limited coverage of traffic detectors makes the travel route unobservant, and ii) there may be multiple paths between two measurement points. We use $R = \{r_1, r_2, ..., r_{|R|}\}$ to denote the alternative paths between two measurement points where $|\cdot|$ is the cardinality of a set. Given an E2E measurement t, we introduce a binary variable $p_{t|r_j}, r_j \in R$ where $p_{t|r_j} = 1$ if t is collected from $r_j \subseteq R$ and $p_{t|r_j} = 0$ otherwise. Obviously, $\sum_{r_j \in R} p_{t|r_j} = 1$ since an E2E measurement is collected only from a unique route. We use $P_{t|R} = \{p_{t|r_1}, p_{t|r_2}, ..., p_{t|r_{|R|}}\}$ to represent the set of binary variables for t based on routes R, so that the probability of t conditioned on $P_{t|R}$ and Θ_R is modeled by:

$$p(t|P_{t|R}, \Theta_R) = \prod_{r_j \in R} p(t_{r_j}|\Theta_{r_j})^{p_{t|r_j}}, \tag{4.8}$$

where $\Theta_R = \{\cup \Theta_{r_j} | r_j \subseteq R\}$. Given the set of E2E measurements between two measurement points in a time interval, T. We define $P_{T|R} = \cup_{t \in T} P_{t|R}$, then the log-likelihood of T is formulated as:

$$\mathcal{L}(T|P_{T|R}, \Theta_R) = \sum_{t \in T} \ln p(t|P_{t|R}, \Theta_R). \tag{4.9}$$

In a road network, suppose we have M pairs of measurement points, then we use $\mathrm{T} = \{t \in T_m | m \in M\}$ to denote the set of all E2E measurements over the whole study site. The log-likelihood of T is formulated as:

$$\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}}) = \textstyle\sum_{m \in M} \mathcal{L}(T_m|P_{T_m|R_m}, \Theta_{R_m}), \tag{4.10}$$

where $\mathrm{R} = \{R_m | m \in M\}$ is the set of the paths with measured data in the road network; $\mathbf{P}_{\mathrm{T|R}} = \cup_{m \in M} P_{T_m|R_m}$ and $\mathbf{\Theta}_{\mathrm{R}} = \cup_{m \in M} \Theta_{R_m}$. By substituting (4.7) and

(4.9) into (4.10), we obtain $\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}})$ as follows:

$$\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}}) = \sum_{m \in M} \sum_{t \in T_m} \sum_{r_j \in R} p_{t|r_j} \ln p(t_{r_j}|\Theta_{r_j}). \tag{4.11}$$

To simplify (4.11), we introduce $\mathbb{R} = \cup_{m \in M} R_m$. Moreover, we define $\mathbf{P}_{\mathrm{T|\mathbb{R}}} = \{p_{t|r_j} | t \in \mathrm{T}, r_j \subseteq \mathbb{R}\}$ where $p_{t|r_j} = 1$ if and only if t is collected on route r_j and otherwise, $p_{t|r_j} = 0$. Obviously, the significance of $\mathbb{R}$ and $\mathbf{P}_{\mathrm{T|\mathbb{R}}}$ are equivalent to R and $P_{\mathrm{T|R}}$. Meanwhile, we introduce $\mathbf{\Theta}_{\mathbb{R}} = \{\cup \Theta_{r_j} | r_j \subseteq \mathbb{R}\}$. As both R and $\mathbb{R}$ should cover all the edges in G, we have $\mathbf{\Theta}_{\mathbb{R}} = \mathbf{\Theta}_{\mathrm{R}} = \{\cup \Theta_{e_k} | e_k \in E\}$. In this case, (4.11) can be represented as

$$\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|\mathbb{R}}}, \mathbf{\Theta}_{\mathbb{R}}) = \sum_{t \in \mathbf{T}} \sum_{r_j \in \mathbb{R}} p_{t|r_j} \ln p(t_{r_j}|\Theta_{r_j}). \tag{4.12}$$

From (4.10), we can observe that the estimation of $\{\mathbf{P}_{\mathrm{T|\mathbb{R}}}, \mathbf{\Theta}_{\mathbb{R}}\}$ relies on $\mathbb{R}$. In the next section, we first illustrate the approaches to estimate $\mathbb{R}$, followed by the estimation of $\{\mathbf{P}_{\mathrm{T|\mathbb{R}}}, \mathbf{\Theta}_{\mathbb{R}}\}$.

4.2 Parameter Estimation

In this section, we estimate $\mathbb{R}$ using a C-shortest paths based algorithm. After that, we estimate $\mathbf{P}_{\mathrm{T|\mathbb{R}}}$ with a K-means algorithm based approach. Next, we estimate $\mathbf{\Theta}_{\mathbb{R}}$ with the EM (Expectation Maximization) algorithm. Finally, to make a trade-off between the complexity and accuracy of the EM algorithm, we design a Q-opt algorithm together with a X-means algorithm.

4.2.1 The Estimation of $\mathbb{R}$

$\mathbb{R}$ has a close relationship with the placement of traffic detectors as well as the road topology. In [129], the authors proposed a traffic camera placement strategy based on the routing matrix W. Particularly, they calculated the bases of W, denoted by $\mathbf{B}_W$, where each basis $B_W \in \mathbf{B}_W$ was defined as a maximal subset of

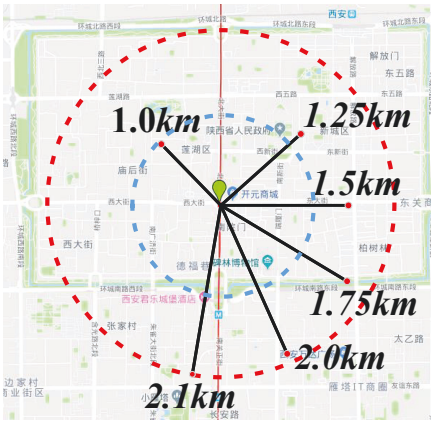
linearly independent routes. After that, the optimal basis $B_{opt} \in \mathbf{B}_W$ was obtained with the minimum cost on the deployment of traffic cameras.

However, the above routing matrix based method is faced with two problems. First, W is usually a high dimensional matrix, especially in the urban road networks. To show the relationship between the number of links and the scale of W, in Fig.4.5a, we select six regions in Xi'an, each of which is a disk centered at Zhonglou with the radius taking a value among $\{1.0\text{km}, 1.25\text{km}, 1.5\text{km}, 1.75\text{km}, 2.0\text{km}, 2.1\text{km}\}$. The larger the radius is, the more links are contained in the region. In Fig.4.5b, the x-coordinate presents the numbers of the links in these six regions are $\{126, 183, 253, 349, 440, 472\}$.

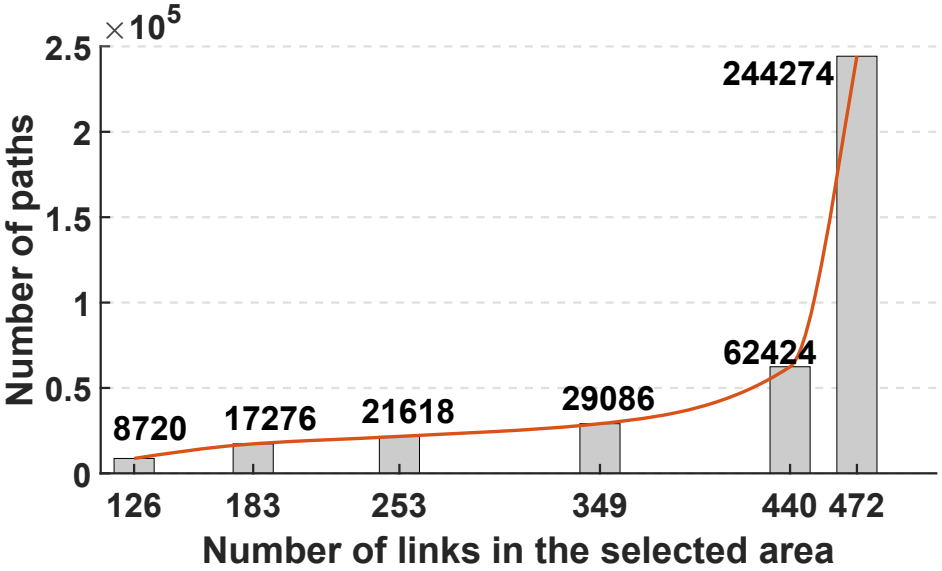
After modeling each region as a digraph G, we present the number of paths in each region in Fig.4.5. Obviously, with the growth of the number of links, the number of rows (paths) in W experiences an approximately exponential growth. The high dimension of W has an negative impact on the estimation of $\mathbf{B}_W$. Such phenomenon is mainly attributable to the fact that there may be hundreds or thousands of paths between two intersections. Second, the route in each row of B_{opt} does not always have observation data. For instance, in Fig.4.6, we present an example where route r_2 (path $V_1 \to V_2 \to V_4 \to V_5 \to V_3$) is obtained from B_{opt}, whereas r_1 (path $V_1 \to V_2 \to V_3$) is achieved with Google Maps. Although r_2 covers more links than r_1, there is no observation data on r_2 since the motorists prefer traveling in r_1 due to the few travel time and short distance.

To solve the above problems, we propose a C-shortest paths based algorithm (Algorithm 4.1) to estimate $\mathbb{R}$. Therein, C is a manually set parameter which is used to control the maximal number of paths between two vertices. From line 2 to line 5, Yen's algorithm [120] is applied to find C-shortest paths between any pair of vertices in V_{neas}, denoted by $\hat{\mathbb{R}}$. The complexity is $O(C|V|^3(|E| + |V|\log|V|))$.

After that, we obtain W, each row of which is a route in $\hat{\mathbb{R}}$ and each column of which is an edge in E. In line 6, a basis B_W of W is estimated using the Bareiss algorithm with the complexity of $O((|V| \cdot |E|)^2)$. Then we estimate $\mathbb{R}$ from line 7 to line 14 with the complexity of $O(|V|^2)$. Clearly, the complexity of Algorithm 4.1 is polynomial.



(a) The selected areas



(b) The number of paths in different selected area

Figure 4.5 : The selected area in Xi'an, China and the number of paths in different areas

Figure 4.6 : A travel route between two intersections using $\mathbf{B}_W$ and Google Maps

4.2.2 The Estimation of $\mathbf{P}_{\mathrm{T}|\mathbb{R}}$

Given an E2E measurement $t \in \mathrm{T}$ detected by the m-th pair of measurement points, we have $p_{t|r_j \subseteq \mathbb{R}-R_m} = 0$ and there exists only one path $r_j \in R_m$ with which $p_{t|r_j \in R_m} = 1$. In this case, The problem of estimating $p_{t|r_j} \in \mathbf{P}_{\mathrm{T}|\mathbb{R}}$ can be interpreted as a clustering problem, that is, to allocate t to a path in $\mathbb{R}$. To this end, we use the K-means algorithm, an unsupervised learning method, which is available for clustering data without labels.

We implement the K-means algorithm on each pair of measurement points parallelly. More precisely, for $T_m \subseteq \mathrm{T}$, we define $K = |R_m|$. Note that a problem we should address is that we do not know which path a cluster represents. As the shorter a path is, the less is the time that a vehicle needs to travel through. We allocate different paths into the clusters using the K-means algorithm in the following way:

- We classify the E2E measurements in T_m into $|R_m|$ clusters with the K-means algorithm.

- We evaluate the average travel time in each cluster and sort the clusters ac-

Algorithm 4.1: The estimation of $\mathbb{R}$

Input: $G = \{V, E\}$, C

Initialization: $\mathbb{R} \leftarrow \varnothing$, $B_W \leftarrow \varnothing$, $\hat{\mathbb{R}} \leftarrow \varnothing$

for $\forall V_i, V_j \in V_{meas}, i \neq j$ **do**

  Estimating R_{ij}^C using Yen's algorithm [120]

  $\hat{\mathbb{R}} \leftarrow \hat{\mathbb{R}} \cup R_{ij}^C$

end

Obtaining W using E and $\hat{\mathbb{R}}$, $B_W \leftarrow$ one basis of W

for $\forall V_i, V_j \in V_{meas}, i \neq j$ **do**

  for *each path* r *in each row of* B_W **do**

    if V_i *and* V_j *are the end points of* r **then**

      $R \cup r$

    end

  end

  $\mathbb{R} \leftarrow \mathbb{R} \cup R$

end

cording to their average travel times.

- We sort the paths according to their lengths, then we map each cluster to each path according to the lengths of paths.

4.2.3 The Estimation of $\mathbf{\Theta}_{\mathbb{R}}$

Recall (4.6), $\forall \Theta_{e_k} \subseteq \mathbf{\Theta}_{\mathbb{R}}$ has the parameters $\{n_{e_k}, h_{e_k}, \mu_{e_k}\}$. Particularly, n_{e_k} is related to the number of E2E measurements collected on the paths that cover e_k. We define a $|\mathbb{R}|$ dimensional vector $P_{t|\mathbb{R}} = (p_{t|r_j} | r_j \subseteq \mathbb{R})$. Then n_{e_k} can be estimated

by

$$n_{e_k} = \sum_{t \in \mathrm{T}} P_{t|\mathbb{R}} \cdot W^k, \tag{4.13}$$

where W is the route matrix estimated using line 6 in Algorithm 4.1, and W^k is the k-th column of W. n_{e_k} is the function of $\mathbf{P}_{\mathrm{T}|\mathbb{R}}$. In this case, the parameters in $\forall\Theta_{e_k} \subseteq \mathbf{\Theta}_{\mathbb{R}}$ are essentially $\{h_{e_k}, \mu_{e_k}\}$.

In order to estimate h_{e_k} and μ_{e_k}, we first simplify the representation of (4.6) based on: 1) the associative property of convolution, that is, $f_1(x)*(f_2(x)+f_3(x)) = f_1(x)*f_2(x)+f_1(x)*f_3(x)$, and 2) the property that the convolution of two Gaussian distributions, i.e. $\mathcal{N}(\mu_1, \sigma_1^2)*\mathcal{N}(\mu_2, \sigma_2^2)$, is also a Gaussian distribution in the format of $\mathcal{N}(\mu_1+\mu_2, \sigma_1^2+\sigma_2^2)$. With these two properties, (4.7) can be rewritten as (4.14) where $\mathcal{Z}_r = \prod_{k=1}^{d_r} n_{e_k}$, $\mu_{r,z} = \sum_{k=1}^{d_r} u_{e_k,i}, \forall i \in n_{e_k}$ and $h_r^2 = \sum_{k=1}^{d_r} h_{e_k}^2$. To better understand (4.14), consider the following case:

$$\begin{aligned} p(\boldsymbol{t}_r|\Theta_r) &= \frac{1}{n_{e_1}h_{e_1}} \sum_{i=1}^{n_{e_1}} \mathcal{N}(\boldsymbol{t}_{e_1}|\mu_{e_1,i}, h_{e_1}^2) * \frac{1}{n_{e_2}h_{e_2}} \sum_{i=1}^{n_{e_2}} \mathcal{N}(\boldsymbol{t}_{e_2}|\mu_{e_2,i}, h_{e_2}^2) * ... * \\ &\quad \frac{1}{n_{e_{d_r}}h_{e_{d_r}}} \sum_{i=1}^{n_{e_{d_r}}} \mathcal{N}(\boldsymbol{t}_{e_{d_r}}|\mu_{e_{d_r},i}, h_{e_{d_r}}^2) \\ &= (\prod_{k=1}^{d_r} \frac{1}{n_{e_k}h_{e_k}}) \cdot \sum_{z=1}^{\mathcal{Z}_r} \mathcal{N}(\boldsymbol{t}_r|\mu_{r,z}, h_r^2), \end{aligned} \tag{4.14}$$

Consider a route r covering two links, each link of which has two E2E measurements, that is, $n_{e_1} = 2$ and $n_{e_2} = 2$. Then, $p(\boldsymbol{t}_r|\Theta_r) = \frac{1}{2h_{e_1}}(\mathcal{N}(\boldsymbol{t}_{e_1}|\mu_{e_1,1}, h_{e_1}^2) + \mathcal{N}(\boldsymbol{t}_{e_1}|\mu_{e_1,2}, h_{e_1}^2)) * \frac{1}{2h_{e_2}}(\mathcal{N}(\boldsymbol{t}_{e_2}|\mu_{e_2,1}, h_{e_2}^2) + \mathcal{N}(\boldsymbol{t}_{e_2}|\mu_{e_2,2}, h_{e_2}^2))$. Based on (4.14), $\mathcal{Z}_r = 2 \times 2$. For each $z \in \mathcal{Z}_r$, we calculate $\mu_{r,z}$ by $\mu_{r,1} = \mu_{e_1,1} + \mu_{e_2,1}$, $\mu_{r,2} = \mu_{e_1,1} + \mu_{e_2,2}$, $\mu_{r,3} = \mu_{e_1,2} + \mu_{e_2,1}$, $\mu_{r,4} = \mu_{e_1,2} + \mu_{e_2,2}$, and calculate $h_{r,z}$ by $h_r^2 = h_{e_1}^2 + h_{e_2}^2$.

Given the natural log of $p(\boldsymbol{t}_r|\Theta_r)$:

$$\ln p(\boldsymbol{t}_r|\Theta_r) = \sum_{k=1}^{d_r} \ln \frac{1}{n_{e_k}} + \sum_{k=1}^{d_r} \ln \frac{1}{h_{e_k}} + \ln \sum_{z=1}^{\mathcal{Z}_r} \mathcal{N}(\boldsymbol{t}_r|\mu_{r,z}, h_r^2). \tag{4.15}$$

we obtain $\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}})$ in (4.16) where $\mathbb{T} = \{t|t \in \mathrm{T}, p_{t|r_j} = 1\}$.

$$\begin{aligned}
\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}}) &= \sum_{t\in\mathbb{T}} \left(\sum_{k=1}^{d_{r_j}} \ln \frac{1}{n_{e_k}} + \sum_{k=1}^{d_{r_j}} \ln \frac{1}{h_{e_k}} + \ln \sum_{z=1}^{\mathcal{Z}_{r_j}} \mathcal{N}(t|\mu_{r_j,z}, h_{r_j}^2) \right) \\
&= \sum_{t\in\mathbb{T}} \sum_{k=1}^{d_{r_j}} \ln \frac{1}{n_{e_k}} + \sum_{t\in\mathbb{T}} \left(\sum_{k=1}^{d_{r_j}} \ln \frac{1}{h_{e_k}} + \ln \sum_{z=1}^{\mathcal{Z}_{r_j}} \mathcal{N}(t|\mu_{r_j,z}, h_{r_j}^2) \right) \\
&= \sum_{t\in\mathbb{T}} \sum_{k=1}^{d_{r_j}} \ln \frac{1}{n_{e_k}} + \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}}), \qquad (4.16)
\end{aligned}$$

From (4.16), we can observe that the parameters $\mathbf{\Theta}_{\mathbb{R}}$ are only included in $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$. Thus, setting the derivative of $\mathcal{L}(\mathbf{T}|\mathbf{P}_{\mathbf{T}|\mathbb{R}}, \mathbf{\Theta}_{\mathbb{R}})$ with respect to $\mathbf{\Theta}_{\mathbb{R}}$ to zero, we have

$$\frac{\mathrm{d}\mathcal{L}(\mathrm{T}|\mathbf{P}_{\mathrm{T|R}}, \mathbf{\Theta}_{\mathrm{R}})}{\mathrm{d}\mathbf{\Theta}_{\mathbb{R}}} = \frac{\mathrm{d}\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\mathrm{d}\mathbf{\Theta}_{\mathbb{R}}} = 0. \qquad (4.17)$$

Unfortunately, there is no closed form solution for (4.17) due to the log of cumulative Gaussian distribution. As a result, the Maximum Likelihood (ML) method does not work here. To address this problem, we employ the EM algorithm to estimate $\mathbf{\Theta}_{\mathbb{R}}$ (Algorithm 4.2) based on the following assumption:

Assumption 1: The $h_{e_k}s$ of the KDE models for the travel time in $\forall e_k \in E$ are same.

To begin with, we introduce the latent variables. For $\forall r_j \subseteq \mathbb{R}$, we define $\mathcal{Z}_{r_j}$-dimensional latent variables as $\boldsymbol{y}_{r_j}$ in which $\forall y_z \in \boldsymbol{y}_{r_j}$ satisfies $y_z = \{0, 1\}$ and $\sum_{y_z\in\boldsymbol{y}_{r_j}} y_z = 1$. Given the definition that the marginal distribution over $\boldsymbol{y}_{r_j}$ is $p(y_z = 1) = \mathcal{Z}_{r_j}^{-1}$, we formulate the distribution of $\boldsymbol{y}_{r_j}$ as $p(\boldsymbol{y}_{r_j}) = \prod_{y_z\in\boldsymbol{y}_{r_j}} \mathcal{Z}_{r_j}^{-y_z}$. We also define the conditional distribution of an E2E measurement t_{r_j} as a Gaussian distribution with $p(t_{r_j}|y_z = 1) = \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2)$. The joint distribution of t_{r_j} is given by:

$$\begin{aligned}
p(t_{r_j}) &= \sum_{\boldsymbol{y}_{r_j}} p(\boldsymbol{y}_{r_j}) p(t_{r_j}|\boldsymbol{y}_{r_j}) \\
&= \mathcal{Z}_{r_j}^{-1} \sum_{z\in\mathcal{Z}_{r_j}} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2) \qquad (4.18)
\end{aligned}$$

We define $\gamma_{r_j}(y_z) \equiv p(y_z = 1|t_{r_j})$, which can be calculated based on Bayes theorem:

$$\begin{aligned} \gamma_{r_j}(y_z) &= \frac{p(y_z = 1)p(t_{r_j}|y_z = 1)}{p(t_{r_j})} \\ &= \frac{\mathcal{N}(t_{r_j}|\mu_{r_j,z}, h^2_{r_j})}{\sum_{z \in \mathcal{Z}_{r_j}} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h^2_{r_j})} \end{aligned} \tag{4.19}$$

Algorithm 4.2: EM algorithm

Input: $\mathbb{R}$

1 **Initialization:** $\mathbf{\Theta}^{(0)}_{\mathbb{R}}$

2 **for** $q \in 1, 2, ...$ **do**

3 **E-step:**

4 $\gamma^{(q)}_{t_{r_j}}(y_z)$: Being updated using (4.19) with $\mathbf{\Theta}^{(q-1)}_{\mathbb{R}}$

5 **M-step:**

6 **for** *each* $\mu^{(q)}_{e_k,i}$ *in* $\Theta^{(q)}_{e_k} \subseteq \mathbf{\Theta}^{(q)}_{\mathbb{R}}$ *and* $h^{(q)}_{e_k}, e_k \in E$ **do**

7 $\mu^{(q)}_{e_k,i} \leftarrow \frac{\sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma^{(q)}_{t_{r_j}}(y_z) t_{r_j}}{N_{r_j}}$

8 $(h^{(q)}_{e_k})^2 \leftarrow \frac{\sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}} \gamma^{(q)}_{t_{r_j}}(y_z)(t_{r_j} - u_{e_k,i})^2}{N_{r_j}}$

9 **end**

10 **Terminal:**

11 **if** $\mathbf{\Theta}^{(q)}_{\mathbb{R}}$ *converges to a local optimum* **then**

12 return $\mathbf{\Theta}^{(q)}_{\mathbb{R}}$

13 **end**

14 **end**

In Algorithm 4.2, $\mathbf{\Theta}^{(0)}_{\mathbb{R}}$ are the initial values of $\mathbf{\Theta}_{\mathbb{R}}$. In line 4, $\mathbb{R}_{e_k} = \{r_j | e_k \in r_j, r_j \subseteq \mathbb{R}\}$, $\mathcal{Z}_{r_j}(u_{e_k,i}) \triangleq \{z | z \in \mathcal{Z}_{r_j}, \mu_{e_k,i} \in u_{r_j,z}\}$ and N_{r_j} is

$$N_{r_j} = \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{r_j}(y_z(t_{r_j})). \tag{4.20}$$

As the performance of the EM algorithm heavily relies on $\mathbf{\Theta}_{\mathbb{R}}^{(0)}$, we use the initialization strategy given in [7]. Convergence is achieved when $\mathbf{\Theta}_{\mathbb{R}}^{(q)} \approx \mathbf{\Theta}_{\mathbb{R}}^{(q-1)}$. The proof of parameters update in line 7 and 8 is presented in the Appendix A.

4.2.4 Q-opt and X-means Based Sampling Algorithm

In the proposed EM algorithm, we can observe that the computational complexity in each iteration depends on $\mathcal{Z}_r$. Further, (4.14) indicates that $\mathcal{Z}_r$ is determined by d_r and n_{e_k}. Therefore, the way to reduce the computational complexity is to limit the path length d_r and reduce the value of n_{e_k}.

Intuitively, the smaller d_r is, the less computational complexity it costs. However, more traffic detectors are needed in the road network if d_r is smaller. Consider the worst case that $d_r = 1$, each path can be viewed as a link. In this case, each intersection should be configured with a traffic detector. To guarantee the accuracy-complexity trade-off, and to control the number of traffic detectors, we propose a Q-opt Algorithm, termed Algorithm 4.3, where Q means the maximal number of links in a path $d_r \leq Q$.

In Algorithm 4.3, $numTD$ is the number of traffic detectors needed in a road network. Line 3 to 8 guarantee the number of paths between two vertices is no more than C and the length of each path is no more than Q. Note that a link may not be covered by any row in $\hat{B}$ (line 11). To address this issue, in line 12 we expand $\hat{B}$ by a $|E|$ dimensional vector I^k where the k-th element in I^k is 1 and the other elements are 0. From line 19 to 22, we obtain B with minimum $numTD$. Given C and Q, the complexity of Algorithm 4.3 is the same as Algorithm 4.1.

n_{e_k}, as the function of $\mathbf{P}_{\mathrm{T|R}}$, is related with $\mathbb{R}$ and T. As $\mathbb{R}$ has been estimated using Algorithm 4.1, the way to reduce n_k is to use the subset of T, denoted by $\hat{\mathrm{T}}$, following the principle that the dynamic characteristics of E2E measurements in T can be perfectly captured by the selected E2E measurements in $\hat{\mathrm{T}}$. To obtain $\hat{\mathrm{T}}$,

Algorithm 4.3: Q-opt

Input: $G = \{V, E\}$, Q

Initialization: $V_{meas} \leftarrow \varnothing$, $numTD$, $\hat{B} \leftarrow \varnothing$

Estimate $\hat{\mathbb{R}}$ using line 2 to 5 in Algorithm 4.1

for $q = 1$ *to* Q **do**

  for *each* $r \subseteq R_{ij}^K, R_{ij}^K \subseteq \hat{\mathbb{R}}$ **do**

    if $d_r > Q$ **then**

      Remove r from R_{ij}^K

    end

  end

  Calculate $\hat{B}$ based on line 6 in Algorithm 4.1

  for *each column* $\hat{B}_k$ **do**

    if $\hat{B}_k = \mathbf{0}$ **then**

      $\hat{B} = \left[\hat{B}; I^k\right]$

    end

  end

  for *each row* $\hat{B}_j$ **do**

    $\boldsymbol{V}_k \leftarrow$the end points of $\hat{B}_j$

    $V_{meas} \cup \boldsymbol{V}_k$

  end

  if $|V_{meas}| < numTD$ **then**

    $numTD \leftarrow |V_{meas}|$

    $B \leftarrow \hat{B}$

  end

end

Estimate $\mathbb{R}$ based on line 7 to 14 in Algorithm 4.1

we first employ the X-means algorithm [68] to classify the E2E measurements on each path $T_{r_j \in \mathbb{R}}$ into X clusters, each of which represents a feature of data. The procedure of the X-means algorithm contains the following three steps:

- Step 1: Given an initial value of $X = X_{in}$, we run the conventional K-means algorihtm until convergence.
- Step 2: To find out whether there is a new centroid using the splitting strategy in [68]. More precisely, we randomly select a centroid and run the K-means. We will accept such a new centroid if the resulting model score is better than before. After that, we have $X = X + 1$.
- Step 3: Repeat the second step until X exceeds a given threshold X_{thre} or there is no improvement on the resulting model score.

Unlike the K-means algorithm where the number of clusters K is manually set in advance, the number of clusters in the X-means algorithm is identified automatically. Thus, the X-means algorithm is able to better capture the dynamic features of E2E measurements in T_{r_j} than the K-means algorithm. After that, we obtain a subset of T_{r_j}, denoted by $\hat{T_{r_j}}$, by selecting data from each cluster using the simple random sampling algorithm [58]. Finally, we obtain $\hat{\mathrm{T}} = \cup \hat{T}_{r_j \subseteq \mathbb{R}}$.

4.3 Experimental Results

4.3.1 Experiment Setup

In this paper, the study site is based on the citywide road network in Xi'an, China, which covers 30,549 links. To validate our proposed method, we use the GPS trajectories anonymously reported by over 11,000 taxicabs on Sep. 5th, 2016 (Mon.). With the average sampling frequency of 30 seconds, we yield over 3.0e+07 raw data records. Each data has the travel information including the time stamp

when the data was sent to the server, the location coordinates (longitude and latitude), the instantaneous travel speed and travel state that takes values from **{stop, cruising, occupied}**. We divide the day into 48 equal time intervals, denoted by $\{\tau_i | i = 1, 2, ..., 48\}$ where $\forall\tau_i$ represents half an hour, e.g., the time interval between 8:00am-8:30am. After that, we set up the model in each τ_i and implement the TTD estimation in Java and Matlab.

Noises exists in the collected GPS data, mainly on account of the precision of GPS. Thus we carry out data preprocessing as follows:

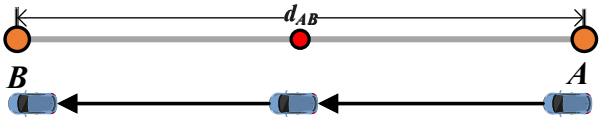
- **Map matching**: We employ a weight-based topological algorithm proposed by Velaga et al. [100]. There are two stages in the the algorithm: i) calculating the weight score for each of the candidate links where a GPS data record is probably in; ii) selecting the link with the highest weight as the correct link for a GPS data record.

- **Outliers filtering**: We filter the outliers mainly including: i) the locations of the GPS data that are out the scope of Xi'an city; ii) the data where the travel speeds exceed the speed limitation, i.e. 120km/h; iii) the data where there are conflicts between the travel state and travel speed (e.g., the state of vehicle is "stop" but the travel speed is not 0); iv) the data where the taxicabs are not in service.

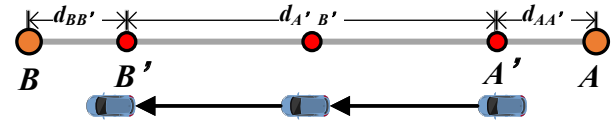
4.3.2 Ground Truth

In this paper, we adopt the following two ground truths: *1) Opt-KDE*: link travel time distributions estimated by KDE with the optimal bandwidth. As discussed in Section 4.1.3, Opt-KDE can fit the distribution of empirical data better than any other models. Thus, we compare the results of our proposed method with Opt-KDE to validate estimation accuracy; *2) Empirical CDF*: the CDF of empirical

data. With KS test defined in Section 4.1.3, we can observe whether the estimated probability distribution is accepted or not.



(a) Only one GPS data record reported by a vehicle



,

(b) More than one GPS data record reported by a vehicle

Figure 4.7 : The instance of calculating link travel time for a vehicle using GPS data

The travel time when a vehicle traverses a link is calculated in two different ways, which depend on the number of GPS data records reported by this vehicle:

- Fig.4.7a is the case with only one GPS data record reported by a vehicle. The travel time t_{AB} is calculated by d_{AB}/v_{AB}, where d_{AB} is the length of link AB and v_{AB} is the space mean speed inferred from the instantaneous speed using the method in Appendix B.

- Fig.4.7b presents the case that multiple GPS data records reported by a vehicle. These data might not exactly reside at the endpoints of the link. In this example, GPS data records are transmitted at the points labeled by red. Assuming A' and B' are the two GPS data records closest to A and B, respectively. Furthermore, the timestamps when the taxicab sent GPS data at A' and B' are $t_{A'}$ and $t_{B'}$. Clearly, $t_{AB} \neq t_{A'B'} = t_{A'} - t_{B'}$. To counter this effect, we apply the method, namely distance and time proportion proposed

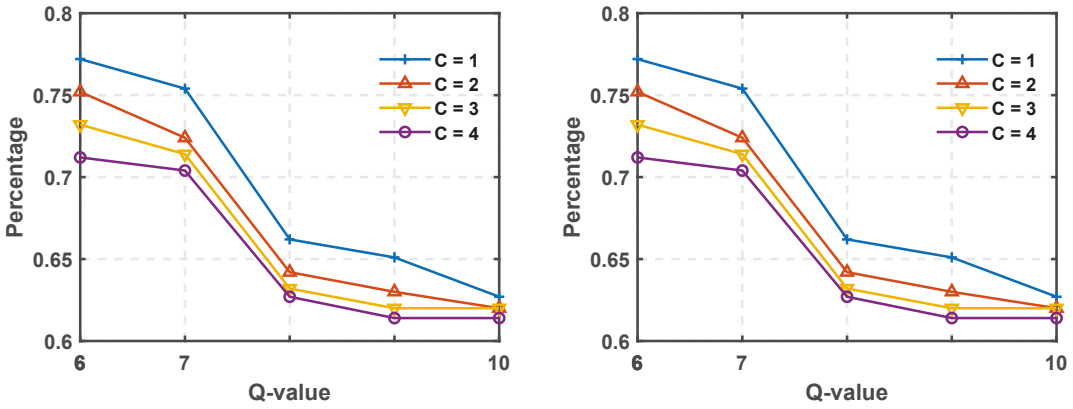
by Sanaullah et al.'s [80], to calculate link travel time. Take Fig.4.7 as an example, t_{AB} is calculated by:

$$t_{AB} = \frac{d_{AB}}{d_{A'B'}} t_{A'B'}. \tag{4.21}$$

Similarly, we use the method to evaluate the path travel times.

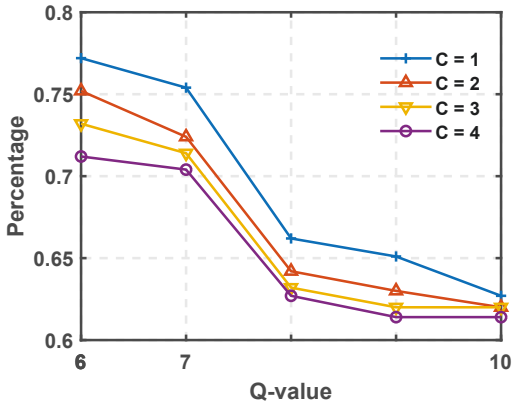
4.3.3 Results

Due to the page limit, we only present the estimated results in the six representative time intervals including the traffic states of free flow and congestion. Note that the parameters of Opt-KDE and other counterpart (including Gaussian, log-normal, GMM, and hazard-based methods) are estimated based on the context that there are complete traffic detectors deployed on the links. In this case, within each time interval, we select a set of links each of which has sufficient observed data (Table 4.3). Compared with these complete detectors based methods, we can also observe the advantage of our proposed method, which only uses a limited number of traffic detectors. In Fig.4.8, we present the percentage of intersections that should deploy traffic detectors. We then observe that fewer traffic detectors are needed when the values of Q and C become larger. In addition, the trend of curves is similar when $C = 2, 3$ and 4. This can be explained by the fact that given a Q, there are at most two candidate paths between most measurement points. Moreover, the percentage converges when $Q = 10$. In the best scenario, approximately 63% of intersections require traffic detectors, and in the worst scenario, approximately 70% of intersections require deploying traffic detectors (in Fig. 4.8f). The convergence reflects the fact that the basis (B_W) obtained from route matrix W changes little when $Q \geq 8$. Based on the above analysis, the following experiments are implemented based on the results obtained from the algorithm with $Q = 10$ and $C = 2$. It is worth noting that the optimal strategy of traffic detector placement is the problem that should be discussed distinctively. It will be studied in our future work.

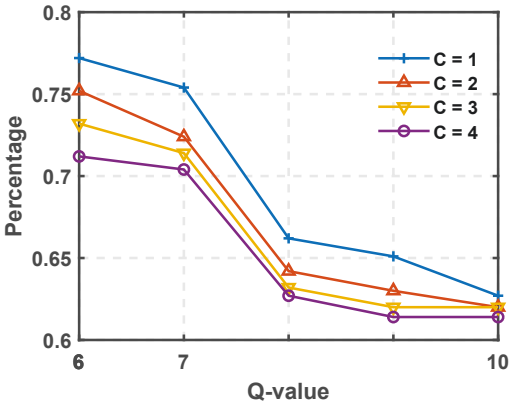


(a) τ_{17}

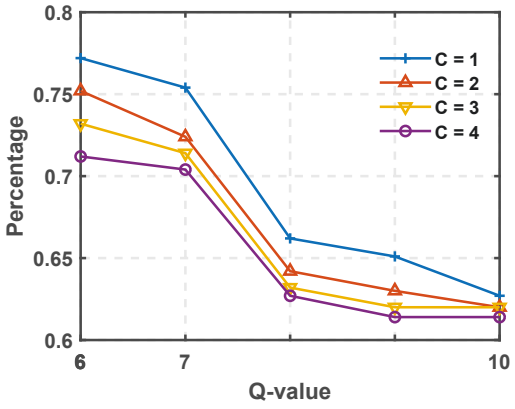
(b) τ_{19}



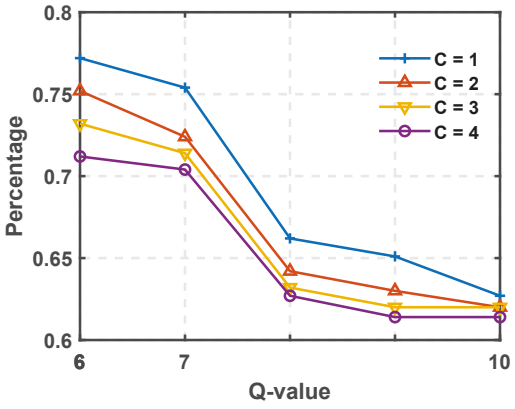
(c) τ_{23}



(d) τ_{31}



(e) τ_{35}



(f) τ_{41}

Figure 4.8 : The percentage of intersections that should deploy traffic detectors with different configurations of Q and C in different time intervals

Table 4.3 : The number of links and travel states in each time interval

	Time intervals	Travel state	No. of links	No. of intersections
τ_{17}	8:00am-8:30am	Congestion	4934	3545
τ_{19}	9:00am-9:30am	Congestion	5271	4031
τ_{23}	11:00am-11:30am	Free flow	5178	3804
τ_{31}	15:00pm-15:30pm	Free flow	5023	3752
τ_{35}	17:00pm-17:30pm	Congestion	5201	3957
τ_{41}	20:00pm-20:30pm	Free flow	4885	3524

Figure 4.9 : The paths between two endpoints A and F

In Table 4.4, we present the experimental results of K-means based algorithm using the data collected by the traffic detectors at the endpoints A and F shown in Fig.4.9. The first path has the links AB, BC, and CF. The second path has the links AB, BD, DE, and EF. In τ_{19}, we have 16 E2E measurements collected on path 1 and 12 on path 2 (the left side of Table 4.4). The right side of Table 4.4 shows that 13 out of 16 (10 out of 12) E2E measurements collected on path 1 (2) are accurately allocated to the corresponding links. We define ACC as clustering accuracy (the percentage of correct decisions). Then, the ACC of K-means for the

Table 4.4 : An instance of K-means based algorithm using the data collected from the paths between A and B in τ_{19}

	Path	No. of data	Path	1	2
1	A-¿B-¿C-¿F	16	1	13	3
2	A-¿B-¿D-¿E-¿F	12	2	2	10

given instance is 82.1%. To present the performance of K-means based algorithm over the whole study site in each time interval, we use the average clustering accuracy ($AACC$) calculated by $AACC = \frac{\sum_{m \in M} ACC_m}{M}$, where ACC_m is the ACC of K-means algorithm implemented on the data collected from the paths between the m-th pair of measurement points. In Table 4.5, we can observe that the best result is obtained in τ_{35} with $AACC = 87.4\%$. The worst result is $AACC = 79.5\%$ in τ_{23}. Compared to the greedy approach used in [129], the experimental results show a good performance of our proposed K-means based algorithm.

Table 4.5 : The $AACC$ of K-means based algorithm and greedy approach

Time interval	τ_{17}	τ_{19}	τ_{23}	τ_{31}	τ_{35}	τ_{41}
K-means	81.2%	80.7%	79.5%	86.2%	87.4%	81.6%
Greedy	74.8%	73.1%	68.3%	71.9%	69.5%	74.7%

Taking path 1 in Fig. 4.9 as an instance, we present the performance of X-means based algorithm in Fig. 4.9. More precisely, the red solid lines show the PDF and CDF of path TTD using Opt-KDE model with 16 E2E measurements. However, only half (8) E2E measurements are needed using X-means based sampling algorithm. From the figures, we can observe that the distributions using the selected E2E measurements is similar with the ones using the whole data. Similar experimental

results are obtained based on the E2E measurements on other paths. Therefore, it is sufficient for us to believe that the E2E measurements filtered by X-means based sampling algorithm can be viewed as the representatives of the whole E2E measurements, and used for parameter estimation without losing too much accuracy. As discussed in Section 4.2.4, with the limited number of E2E measurements, the efficiency of EM algorithm will be improved.

Table 4.6 : Average KL divergence for different models in the selected time intervals

Time interval	KDE-E2E	Gaussian	log-normal	GMM	Hazard
τ_{17}	0.92	1.51	1.24	0.93	1.13
τ_{19}	**1.03**	1.46	1.37	**0.89**	1.07
τ_{23}	0.96	1.73	1.19	1.01	1.15
τ_{31}	**0.90**	1.93	0.91	**0.87**	0.97
τ_{35}	0.94	1.49	1.25	0.98	1.14
τ_{41}	0.86	1.28	0.94	0.91	1.02

To assess the deviation between an estimated TTD and the ground truth (Opt-KDE) in a link, we use the metric named Kullback Leibler (KL) divergence, which is defined as follows:

$$D_{KL}(P_{opt}||P_{es}) = \sum_{t \in T_{e_k}} p_{em}(t) \ln \frac{p_{em}(t)}{p_{es}(t)}, e_k \in E. \tag{4.22}$$

In (4.22), p_{opt} represents the TTD of Opt-KDE, and p_{es} is the estimated TTD with our proposed model (namely KDE-E2E) and its counterparts. In particular, GMM has three components of Gaussians. The hazard-based model was proposed by Emily and Taha in [63]. It has a good performance in estimating TTD by considering the factors like travel speed, weather, road condition, etc. As we do not have the data

like the weather, we cannot re-build the model as the one in [63]. In our paper, we only use the traffic speed and road condition to set up the hazard-based model. To evaluate the performance of the estimated results, we define the average KL divergence by:

$$\bar{D}_{KL}(P_{opt}||P_{es}) = \frac{\sum_{e_k \in E} D_{KL}(P_{opt}||P_{es})}{|E|}. \tag{4.23}$$

Table 4.8 : Average KL divergence and KS test over 48 time intervals

Models	KDE-E2E	Gaussian	log-normal	GMM	Hazard
KL	0.89	1.62	1.31	0.93	1.07
KS	0	37	25	0	5

Table 4.9 : Average KL divergence and KS test over all the links in each time interval

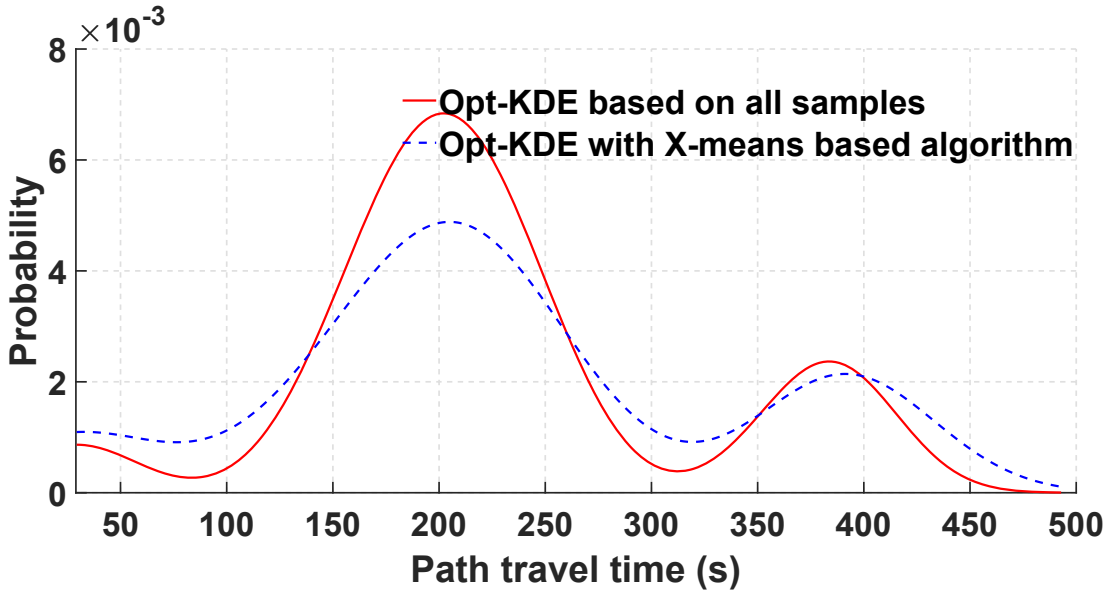
Time interval	Gaussian		log-normal		GMM		Hazard	
	KL	KS	KL	KS	KL	KS	KL	KS
τ_{17}	100%	86.2%	99.8%	54.8%	13.5%	0.5%	99.1%	3.4%
τ_{19}	100%	90.4%	98.3%	58.5%	4.6%	0	97.0%	5.3%
τ_{23}	100%	83.2%	100%	63.0%	9.2%	0.3%	99.5%	4.7%
τ_{31}	100%	75.9%	100%	55.3%	6.9%	0	100%	3.4%
τ_{35}	100%	92.2%	99.4%	58.3%	12.1%	0.3%	98.6%	4.5%
τ_{41}	100%	80.4%	100%	61.4%	14.3%	0.8%	99.7%	9.2%

From Table 4.6, we can observe that the performance of KDE-E2E is always better than Gaussian, log-normal and hazard-based model in each time interval, but a little worse than GMM in τ_{17} and τ_{31}. This can be explained by the fact

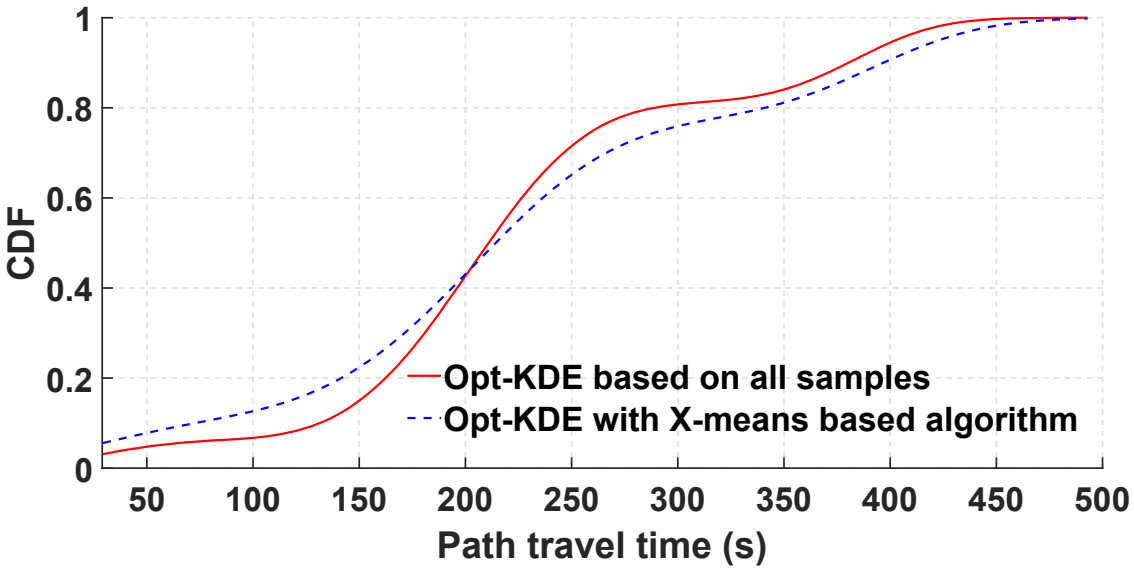
Table 4.7 : The KS test based on different probabilistic models with significance level $\alpha = 0.01$

$D_{em,es}$	τ_{17} $c(\alpha) = 0.120$	τ_{19} $c(\alpha) = 0.085$	τ_{23} $c(\alpha) = 0.093$	τ_{31} $c(\alpha) = 0.115$	τ_{35} $c(\alpha) = 0.960$	τ_{41} $c(\alpha) = 0.120$
Opt-KDE	0.027	0.033	0.021	0.029	0.045	0.030
KDE-E2E	0.042	**0.059**	0.037	0.045	0.064	0.038
Gaussian	0.231 (reject)	0.189 (reject)	0.204 (reject)	0.176 (reject)	0.271 (reject)	0.266 (reject)
log-normal	0.117	0.134 (reject)	0.802	0.095	0.131 (reject)	0.128 (reject)
GMM	0.051	**0.046**	0.043	0.055	0.088	0.039
Hazard	0.109	0.087	0.091	0.103	0.882	0.130 (reject)

that GMM has the similar structure with Opt-KDE. However, in the real world, it is difficult to estimate the parameters of GMM because there is usually a lack of data on the target links. By comparing $\bar{D}_{KL}$s of KDE-E2E under different road conditions, we can also find out that the minimal and maximal $\bar{D}_{KL}$s under the free flow are 0.86 (τ_{41}) and 0.96 (τ_{23}), whereas the minimal and maximal $\bar{D}_{KL}$s under the congestion are 0.92 (τ_{17}) and 1.03 (τ_{19}). The smaller is$\bar{D}_{KL}$, the better is the estimation accuracy. Therefore, the estimation accuracy of our proposed KDE-E2E method is superior under the free flow. It can be explained by the fact that the fluctuation of travel times is usually large under the congestion. Thus, it is difficult to capture all the features of the variation of travel times.



(a) PDFs



(d) CDFs

Figure 4.9 : The performance of X-means based algorithm based on the instance in Fig.4.9

In Table 4.7, we use the KS test to measure the similarity between the empirical CDF and the estimated one based on our proposed model and the counterparts. Particularly, we present the results of a randomly selected link. Obviously, our proposed model is accepted in each time interval. Compared to KDE-E2E and GMM, we can find that $D_{em,es}$ of GMM in τ_{19} is smaller than $D_{em,es}$ obtained from our proposed model. This result is consistent with the result in Table 4.6.

In Table 4.8, we calculate KL divergence and implement KS test over 48 time intervals based on KDE-E2E and its counterparts. The second row denotes average KL divergence ($\bar{D}_{KL}$) and the third row denotes the number of time intervals in which the model is rejected. From Table 4.8, we can observe that $\bar{D}_{KL}$s of KDE-E2E are smaller than the other models over 48 time intervals. Meanwhile, KDE-E2E models are all accepted in the whole time intervals. The second best model is GMM. Not surprisingly, the Gaussian model has the worst performance since it is rejected in

37 time intervals, whereas our proposed methods are accepted in each time interval. In Table 4.9, we compare KDE-E2E with other models with the same metrics over all the links in each time interval. More precisely, the column labeled by KL shows the percentage of links that our model is better than the other methods based on KL divergence. The column labeled by KS means the percentage of the links that our model is better than the other models based on KS test. Obviously, our method is much better than Gaussian and log-normal models. According to KS test results, GMM and the hazard-based model have good performance for link TTD estimation. However, $\bar{D}_{KL}$s of hazard-based model are still smaller than those obtained by our proposed KDE-E2E over most links. As for GMM, there are still at most 14.3% links whose $\bar{D}_{KL}s$ are smaller than our method. This relies on the fact that the fixed number of Gaussian components in GMM has the limitation of capturing all the features of TTD. Combining with experimental results in Table 4.6 and 4.7, the efficiency of our proposed method is further validated.

4.4 Summary

Motivated by the network tomography, in this chapter, we estimated TTDs with the E2E measurements detected by a limited number of traffic detectors deployed at or near the intersections. With the proposed KDE-E2E method, traffic administrators can deploy traffic detectors (e.g., traffic cameras) or dispatch probe vehicles to collect traffic data at some critical positions. Thus, a lot of resources can be saved. Also, through observing and analyzing the distribution of travel times in the links, traffic administrators can carry out effective measures to avoid the occurrence of congestions.

Chapter 5

Graph Neural Network And Distributed Lagrange Dual Decomposition Based Method For Taxi Cruising Route Recommendation

Taxi cruising route recommendation has the benefit of reducing idle time of taxis and waiting time of passengers. However, existing methods still suffer from some shortcomings. First, they primarily focus on cruising route recommendation for a single taxi with less consideration of potential competition and collaboration between the taxis. As a result, the solution easily falls into local optimum. Second, most existing methods are centralized approaches, which consume a large amount of computational resource. Third, the inputs of existing predictors of taxi demands are usually Euclidean data, which has the limitation of capturing the spatiotemporal correlation between taxi demands. To cope with these challenges, in this chapter, we formulate taxi recommendation as a bi-objective optimization problem with the aim of minimizing the number of vacant taxis to gain individual revenue and maximizing global revenue by considering collaboration and competition between taxis. We first partition the urban network using a joint SLPA and GN algorithm. After that, taxi demand in each sub-area is forecasted using the proposed LSTM-GCN model. Finally, we obtain the solution to the problem using the proposed distributed algorithm based on Lagrange dual decomposition.

The organization of the paper is as follows: The problem formulation is described in Section 5.1. The graph partition algorithm, taxi demand and destination predictors, and the distributed optimization algorithm are introduced in Section 5.2. The performance of our proposed method is validated and investigated in Section 5.3.

Finally, we draw the summary in Section 5.4.

5.1 Problem Formulation

Table 5.1 : Major Symbols

Symbols	Significance
$G = \{U, E\}$	Digraph model of the road network. Each vertex $u_i \in U$ is a link. There is a directed edge $e_j \in E$ from u_1 to u_2 if there is traffic traveling from u_1 to u_2
r	A route or a trip is composed of a series of vertices and directed edges
$r^{\mathcal{C}}$	A route from current location of taxi to a pick-up location of a passenger
$r^{\mathcal{O}}$	A route from a pick-up location to a destination
h/z	Pick-up zone/Drop-off zone
H/Z	A set of pick-up/drop-off zones
$\mathcal{C}_{nm}^{h_i}$	A set of cruising routes for the n-th taxi that picks up the m-th passenger from a pick-up zone $h_i \in H$
$\mathcal{O}_{nm}^{z_j}$	A set of occupied routes for the n-th taxi that drops off the m-th passenger in a destination zone $z_j \in Z$
τ	Time interval, e.g., 1 hour
$t - \omega\tau$	The ω-th time interval $((t - \omega\tau, t - (\omega - 1)\tau])$ before recent time t
N_t	Number of vacant taxis at t
$M_{t-\omega\tau}$	The number of taxi demand within the ω-th time interval
$y_{h_i,t-\omega\tau}$	Taxi demands in $h_i \in H$ within the ω-th time interval
$\boldsymbol{y}_{t-\omega\tau}$	$\boldsymbol{y}_{t-\omega\tau} = \cup_{h_i \in H} y_{h_i,t-\omega\tau}$
$\boldsymbol{x}_{t-\tau}$	$\boldsymbol{x}_{t-\tau} = \{x_{nm} \mid n \in N_t, m \in M_{t-\tau}\}$ where $x_{nm} = 0$ or 1

Continued on next page

Table 5.1 continued from previous page

Symbols	Significance
$p(r^{\mathcal{C}}_{nm})/\ p(r^{\mathcal{O}}_{nm})$	Probability for the n-th taxi traveling through $r^{\mathcal{C}}_{nm}/r^{\mathcal{O}}_{nm}$
$R(\cdot)$	Revenue function
$p(z_j\|h_i, t-\omega\tau)$	Probability for a passenger traveling from h_i to z_j within the ω–th time interval
$\mathrm{P}(Z\|H, t-\omega\tau)$	$\{p(z_j\|h_i, t-\omega\tau)\|\forall z_j \in Z, \forall h_i \in H\}$
w_1, w_2	Weights for bi-objective
sf_1, sf_2	Scale factors of bi-objective
K	Number of POIs attached to a link (vertex)
$\mathcal{T}$	Number of iterations
ϵ_{thr}	Maximal number of communities of a vertex
ϵ_{size}	Maximal size of a community
$\mathcal{G} = \{H, \mathcal{E}\}$	Undirected graph. Each vertex$h_i \in H$ is a pick-up zone. There is an undirected edge $\varepsilon_j \in \mathcal{E}$ between h_{i_1} and h_{i_2} if there are at least two links respectively in h_{i_1} and h_{i_2} connecting with each other
L	Normalized Laplacian matrix of $\mathcal{G}$, $L = Q\Lambda Q$
Q	A matrix of eigenvectors of L
Λ	Diagonal matrix. Each diagonal dlement is an eignvalue of L
$*_{\mathcal{G}}$	Graph convolution
κ	Graph convolution kernel
Θ	Chevyshev polynomial kernel
$\boldsymbol{f}/\boldsymbol{i}/\boldsymbol{o}$	Forget/Input/Output gate
$\widehat{\boldsymbol{H}_\tau}/\boldsymbol{H}_\tau$	Intermediate hidden state/Hidden state

Continued on next page

Table 5.1 continued from previous page

Symbols	Significance
W_{hr}/W_{day}	Weights for the predicted results from two sub-networks in the proposed GCN based model

To help readers keep track of symbols' meanings, we clarify the major notations in Table 5.1. Before problem formulation, we first model the road network as an undirected graph $G = \{U, E\}$ where U is a set of vertices and E is a set of directed edges. Each vertex $u \in U$ represents a link defined as a one-way road segment bounded by two adjacent road intersections. There exists an edge $e \in E, e = (u_1, u_2)$ if link u_1 and u_2 are connected. A route r is a sequence of vertices $u_1, u_2, ..., u_{l_r}$ and edges $(u_1, u_2), (u_2, u_3), ..., (u_{|r|-1}, u_{|r|})$ where $|r|$ is the number of links in r. With the graph model, we define some notions which will be used in the problem formulation.

Definition 1: Pick-up/Drop-off zone: With the knowledge of road topology and geographical distribution of POIs, we divide G into a set of disjoint sub-graphs where each sub-graph is viewed as a pick-up zone (denoted by h) or a destination zone (denoted by z). After graph partition, the links clustered into the same zone are close to each other or close to the same POI like a shopping mall.

With Definition 1 and graph model G, we will obtain a set of pick-up and drop-off zones, denoted by H and Z. The approach to partitioning G into H and Z will be illustrated in **Section 5.2.1**. Consider the fact that the passengers usually wait for the vacant taxis by the side of a link, we assume a pick-up or a drop-off location is **in the middle of a link** in $h_i \in H$ or $z_j \in Z$. Further, we define a cruising/occupied route as follows:

Definition 2: Cruising/Occupied route: A route that the taxis travel from current/pick-up location to pick-up/drop-off location, respectively denoted by $r^{\mathcal{C}}$ and $r^{\mathcal{O}}$.

In practice, there may be multiple cruising or occupied routes between two locations. For simplicity, in this paper, we only consider a **unique** route, which usually has the shortest distance or the least travel time. We use τ to denote a time interval, e.g., one hour. The vacant taxis at recent time t is N_t which can be obtained by the tracing devices equipped in the taxis. Besides, we denote taxi demand in a pick-up zone $h_i \in H$ within a time interval $(t-\omega\tau, t-(\omega-1)\tau]$ by $y_{h_i,t-\omega\tau}$. $y_{h_i,t-\omega\tau}$ represents the taxi demand in the ω-th time interval before t if $\omega > 0$, whereas in the ω-th time interval after t if $\omega < 0$. Then, we denote the taxi demand in the whole road network by $\boldsymbol{y}_{t-\omega\tau} = \{y_{h_i,t-\omega\tau} | h_i \in H\}$, which is a $\mathbb{R}^{|H|\times 1}$ vector. The number of taxi demand is calculated as $M_{t-\omega t} = \sum_{h_i \in H} |y_{h_i,t-\omega\tau}|$. The prediction of $\boldsymbol{y}_{t-\omega\tau}, \omega < 0$ will be illustrated in **Section 5.2.2**. We introduce a set of binary variables $\boldsymbol{x}_{t+\tau} = \{x_{nm} | n \in N_t, m \in M_{t+\tau}\}$ where $x_{nm} = 1$ if the n-th vacant taxi is allocated to the m-th passenger, otherwise, $x_{nm} = 0$. Then the problem is formulated as follows:

$$\mathbf{P}: \quad \{\min\ f_1(\boldsymbol{x}_{t+\tau}), \max\ f_2(\boldsymbol{x}_{t+\tau})\} \tag{5.1}$$

$$s.t. \quad \sum_{m \in M_{t+\tau}} x_{nm} \in \{0,1\}, \forall n \in N_t \tag{5.2}$$

$$\sum_{n \in N_t} x_{nm} \in \{0,1\}, \forall m \in M_{t+\tau} \tag{5.3}$$

$$\sum_{m \in M_{t+\tau}} R(x_{nm}) \geqslant \epsilon_{rev}, \forall n \in N_t \tag{5.4}$$

$$x_{nm} \in \{0,1\}, \forall n \in N_t, m \in M_{t+\tau} \tag{5.5}$$

Specifically, constraint (5.2) and (5.3) guarantee that a taxi can only pick up one passenger at the same time, and vice versa. Constraint (5.4) guarantees the revenue of the n-th taxi is above an constant expected revenue, ϵ_{rev}. Lastly, constraint (5.5) indicates x_{nm} is a binary variable. There are two objectives in the problem $\mathbf{P}$, respectively minimizing the number of vacant taxis ($f_1(\boldsymbol{x}_{t+\tau})$), and maximizing the global revenue of all the vacant taxis ($f_2(\boldsymbol{x}_{t+\tau})$). More precisely, $f_1(\boldsymbol{x}_{t+\tau})$ is

expressed by

$$f_1(\boldsymbol{x}_{t+\tau}) = N_t - \sum_{n \in N_t} \sum_{m \in M_{t+\tau}} x_{nm}, \tag{5.6}$$

and $f_1(\boldsymbol{x}_{t+\tau}) \geqslant 0$ due to constraint (5.2). $f_2(\boldsymbol{x}_{t+\tau})$ is expressed by

$$f_2(\boldsymbol{x}_{t+\tau}) = \sum_{n \in N_t} \sum_{m \in M_{t+\tau}} R(x_{nm}), \tag{5.7}$$

where $R(x_{nm})$ is the revenue function. Obviously, $R(x_{nm}) = 0$ if $x_{nm} = 0$, otherwise, suppose the m-th passenger allocated to the n-th taxi is from $h_i \in H$, then $R(x_{nm})$ depends on the fare and cost over the trip $r_{nm} = \{r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm}\}$. Note that the pick-up location of the m-th passenger could be any link in h_i, thus, for the m-th passenger, there would be $|h_i|$ possible $r^{\mathcal{C}}_{nm}$ for the n-th taxi, denoted by $\mathcal{C}^{h_i}_{nm}$ where the end-points of each cruising route $r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}$ correspond to the current location of the n-th taxi and a possible pick-up location for the m-th passenger. Similarly, given a drop-off zone z_j for the m-th passenger, we define $\mathcal{O}^{z_j}_{nm}$ as a set of possible occupied routes with $|\mathcal{O}^{z_j}_{nm}| = |z_j|$. After that, we calcuate the cost over r_{nm} conditioned on $r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}$ and $r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}$ by

$$cost(r_{nm}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm}) = c(t_{r^{\mathcal{C}}_{nm}} + t_{r^{\mathcal{O}}_{nm}}). \tag{5.8}$$

where c is the fuel cost per kilometer, t_r is the travel time upon r and estimated by Google Maps API *. In this paper, we formulate the cost as the function of travel time as opposed to the length of the route due to the following reasons: 1) with the fixed travel speed, the longer the distance is, the larger the cost is; 2) with the fixed travel distance, the smaller the travel speed is, the longer the waiting time is, and the larger the cost will be; 3) travel time can reflect the impact of different road conditions on the decision making from taxi drivers. A taxi driver likely dismisses a passenger's request due to the congestion happening on a cruising route.

*Google Maps API is called by javascript.

We use $p(r^{\mathcal{C}}_{nm})$ and $p(r^{\mathcal{O}}_{nm})$ to denote the probabilities that a taxi travels through $r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}$ and $r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}$, respectively. Further, we use $p(z_j|h_i, t+\tau)$ to denote the probability that a taxi travels to $z_j \in Z$ from $h_i \in H$ within the time interval $(t, t+\tau]$. Particularly, $\sum_{r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}} p(r^{\mathcal{C}}_{nm}) = 1$ and $\sum_{z_j \in Z} p(z_j|h_i, t+\tau) \sum_{r^{\mathcal{C}}_{nm} \in \mathcal{O}^{z_j}_{nm}} p(r^{\mathcal{O}}_{nm}) = 1$. In this case, the cost on the trip r_{nm} is the conditional expectation of $cost(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm})$, which calculated by

$$\begin{aligned} \mathbf{E}(cost(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm})) = c(& \sum_{r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}} p(r^{\mathcal{C}}_{nm}) t_{r^{\mathcal{C}}_{nm}} \\ & + \sum_{z_j \in Z} p(z_j|h_i, t+\tau) \sum_{r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}} p(r^{\mathcal{O}}_{nm}) t_{r^{\mathcal{O}}_{nm}}). \end{aligned} \tag{5.9}$$

According to Definition 1, the links in a pick-up zone h_i are close to each other or close to the same POI. Therefore, it is reasonable to assume that the probability for a passenger waiting for a taxi by the side of any link of h_i follows the uniform distribution, that is, $p(r^{\mathcal{C}}_{nm}) = \frac{1}{|\mathcal{C}^{h_i}_{nm}|}, r^{\mathcal{C}}_{nm} \in \mathcal{C}^{h_i}_{nm}$. Similarly, $p(r^{\mathcal{O}}_{nm}) = \frac{1}{|\mathcal{O}^{z_j}_{nm}|}, r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}$. The estimation of $p(z_j|h_i, t+\tau)$ is more complicated than $p(r^{\mathcal{C}}_{nm})$ or $p(r^{\mathcal{O}}_{nm})$ due to the following two reasons. First, the destination of a passenger heavily depends on the pick-up location where the passenger leaves. Second, the destination of a passenger depends on his/her behavior which, in most cases, follows a specific pattern. For instance, in the morning of weekday, a passenger from a pick-up zone such as residential area more likely goes to the destination zone such as Central Business District (CBD). We define $\mathrm{P}(Z|H, t+\tau) = \{p(z_j|h_i, t+\tau)|\forall z_j \in Z, \forall h_i \in H\}$ which will be illustrated in **Section 5.2.3**.

The fare over r_{nm} conditioned on $r^{\mathcal{C}}_{nm}$ and $r^{\mathcal{O}}_{nm}$ is calculated by:

$$fare(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm}) = f t_{r^{\mathcal{O}}_{nm}}, \tag{5.10}$$

where f is the fare per kilometer. The fare over the trip r_{nm} is the conditional

expectation of $fare(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm})$, which is calculated by

$$\begin{aligned} &\mathbf{E}(fare(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm})) = \\ f \sum_{z_j \in Z} p(z_j|h_i, t+\tau) &\sum_{r^{\mathcal{O}}_{nm} \in \mathcal{O}^{z_j}_{nm}} p(r^{\mathcal{O}}_{nm}) t_{r^{\mathcal{O}}_{nm}}. \end{aligned} \tag{5.11}$$

Thus, $R(x_{nm})$ with $x_{nm} = 1$ is estimated by:

$$\begin{aligned} R(x_{nm} = 1) &= \mathbf{E}(cost(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm})) \\ &+ \mathbf{E}(fare(r_{mn}|r^{\mathcal{C}}_{nm}, r^{\mathcal{O}}_{nm})). \end{aligned} \tag{5.12}$$

We formulate two objectives in $\mathbf{P}$ since the optimal solution obtained from single objective based formulation, e.g., $f_2(\boldsymbol{x}_{t+\tau})$, may lead to some shortcomings, e.g., more vacant taxis. To illustrate, consider the instance in Table 5.2. On the left side of Table 5.2, we present an instance including three passengers $\{A, B, C\}$ and three vacant taxis $\{a, b, c\}$. Each element is the revenue of a taxi corresponding to a passenger. With the single objective $f_2(\boldsymbol{x}_{t+\tau})$, the optimal solution is shown on the top right of Table 5.2 where taxi c is not allocated to any passenger. In practice, each taxi driver wants to get a passenger to gain his/her own revenue without considering the other taxis. Therefore, in our problem formulation, we aim at minimizing the vacant taxis in the road network to improve the individual revenue of taxis and maximizing global revenue of the whole taxis. With bi-objective based formulation, the optimal solution to the same problem in Table 5.2 is presented on the bottom right. Although global revenue is reduced to 18, all the vacant taxis can get a passenger.

To solve $\mathbf{P}$, we first rewrite the bi-objective by $\min\{f_1(\boldsymbol{x}_{t+\tau}), -f_2(\boldsymbol{x}_{t+\tau})\}$. Then we use the weighted sum method [56] to combine all the objective functions to a single function. In this case, $\mathbf{P}$ can be transformed to $\mathbf{P}'$, denoted by

$$\begin{aligned} \mathbf{P}': \quad & \min \ \tfrac{w_1}{sf_1} f_1(\boldsymbol{x}_{t+\tau}) - \tfrac{w_2}{sf_2} f_2(\boldsymbol{x}_{t+\tau}) \\ s.t. \quad & (5.2), (5.3), (5.4), (5.5) \end{aligned} \tag{5.13}$$

Table 5.2 : An instance of two solutions respectively obtained from single objective based and bi-objective based problem formulations

	A	B	C	$f_2(\boldsymbol{x}_{t+\tau})$ based formulation
a	12	3	-6	Solution: a:A, b:B; Revenue:19
b	2	7	4	$f_1(\boldsymbol{x}_{t+\tau})$,$f_2(\boldsymbol{x}_{t+\tau})$ based formulation
c	4	2	-5	Solution: a:A, b:C, c:B; Revenue:18

where sf_1, sf_2 are the scale factors used for normalizing the bi-objective, and $w_1, w_2 > 0$ are the weights with $w_1 + w_2 = 1$. In **Section 5.2.4**, we propose a distributed algorithm to obtain the solution to $\mathbf{P}'$.

5.2 Solution

In this section, we first estimate H and Z with SLPA and GN algorithm, namely GN-SLPA. Then we predict taxi demand in next time interval of current time, $\boldsymbol{y}_{t+\tau}$ using the proposed GCN based method. After that, we estimate $\mathrm{P}(Z|H, t+\tau)$ with a statistical model. Lastly, the solution of $\mathbf{P}'$ is obtained with the distributed dual ascent algorithm.

5.2.1 The Estimation of H and Z

Conventional solutions to partition a road network mainly include uniform (eg. regular grid layout [101, 119]) and non-uniform methods (e.g., quadtree and k-dimensional tree based schemes [92]). In these methods, the map of an urban area is divided into a group of zones where the shape of each zone is regular like rectengle and circle, whereas non-regular after partitioning by non-uniform methods. Different from these methods,in this paper, our proposed partition algorithm is based on the graph model of the urban road network. After partition, we get a set of sub-graphs

where each sub-graph, also known as community, is a pick-up or a drop-off zone based on Definition 1.

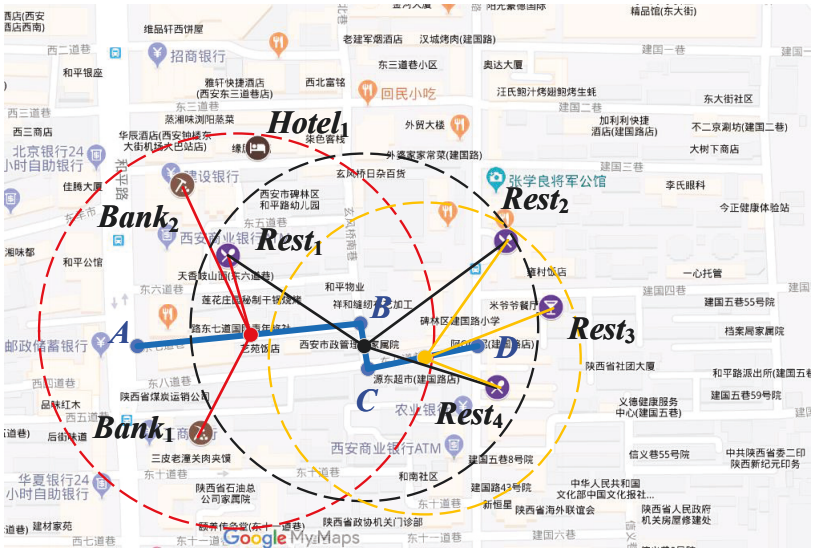


Figure 5.1 : The way to determine K POIs around a link.

Suppose we have a series of POIs where each POI is associated with an identifier like shopping mall or restaurant. Given a vertex $u_i \in U$, we take the middle of the i-th link (u_i) as the center and get a circle in which there are K' POIs. After that, we order K' POIs according to the distance between u_i and each POI and attach u_i with $K \leqslant K'$ nearest POIs. Lastly, GN-SLPA [115] is implemented with the following steps:

Step 1: Each vertex has a memory used for storing the information propagated from other vertices. Also, the memory of each vertex is initialized with the community identifiers of its K nearest POIs.

Step 2: A vertex u_i is selected as a listener and its neighbors are viewed as speakers. Each speaker randomly selects a community identifier with the belonging coefficient proportional to its frequency in the memory of the speaker and then propagates such community identifier to v_i. After receiving a series of community

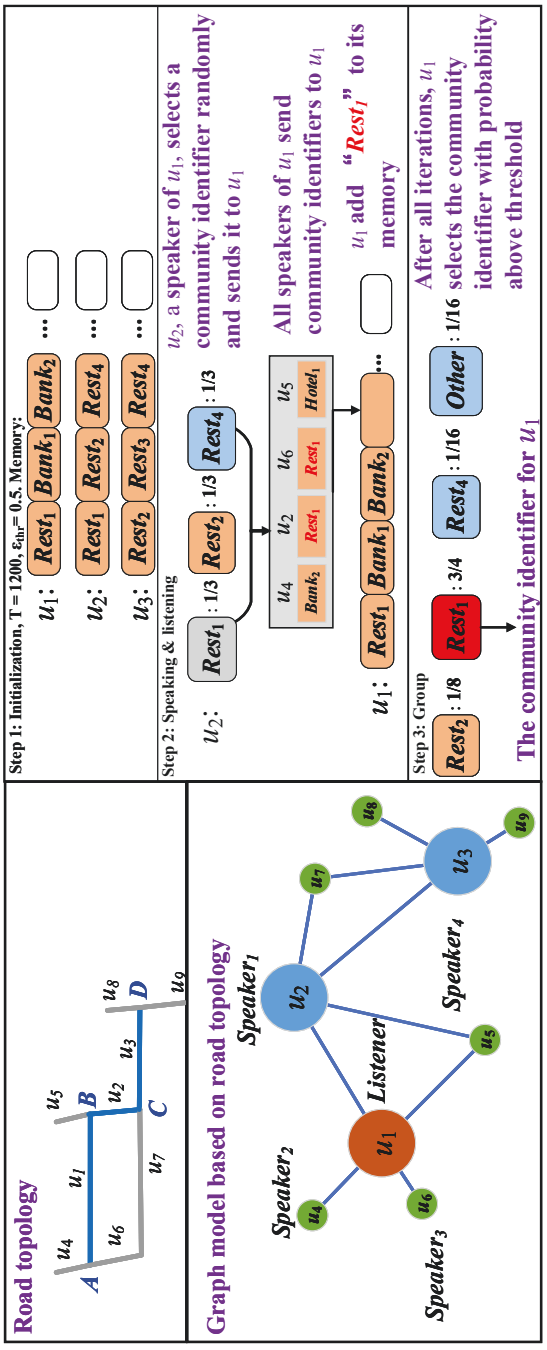


Figure 5.2 : An instance of SLPA algorithm based on the road topology in Fig. 5.1.

identifiers, the listener stores the most common one into its memory.

Step 3: **Step 2** is repeated $\mathcal{T}$ (1200 in this paper, see Section (5.2.1)) iterations until the result of partition converge. Finally, we build a probability distribution of community identifiers stored in u_i's memory. Given a threshold $\epsilon_{thr} \in [0, 1]$, the community identifier with the probability less than ϵ_{thr} will be deleted. The vertices with common community identifiers are then grouped into a community. SLPA is an overlapping algorithm with which a vertex may belong to multiple communities. However, in our paper, we aim at partitioning a graph into a set of disjoint communities. In this case, we set $\epsilon_{thr} \geqslant 0.5$ [37]. To better understand the process of SLPA algorithm, an instance is presented in Fig.5.1 and 5.2. Particularly, Fig.5.1 illustrates the way to select K POIs of a link. In this instance, there are three links $\{AB, BC, CD\}$, which are modeled as vertices by $\{u_1, u_2, u_3\}$. Given $K = 3$ and radius = 0.2km, u_1 has 4 POIs. We order them based on the distance between each POI and u_1 by $\{Rest1, 0.27km\}, \{Bank1, 0.28km\}, \{Bank2, 0.3km\}, \{Hotel1, 0.32km\}$. Then, the nearest 3 POIs are associated with u_1. Fig.5.2 shows the instance of grouping a link into a community with the three steps.

Step 4: Given a community with size (the number of vertices) larger than ϵ_{size} (120 in this paper), we utilize GN algorithm to divide such community into a series of sub-communities with small sizes. The core idea of GN algorithm is as follows. First, the betweenness [†] of all existing edges in the network is calculated. Second, the edge with the highest betweenness is removed, and the betweenness of all edges after the removal will be recalculated. The above two procedures are repeated until no edges remain.

The purpose of Step 4 is to limit the size of each community in order to avoid

[†]Given a vertex $u_i \in U$, the betweenness associated with u_i is defined as the number of shortest paths between pairs of nodes that run through u_i

the case that the pick-up and drop-off location of a passenger are located in the same zone, that is, $h_i = z_j$. In the situation of $h_i = z_j$, $p(z_j|h_i, t+\tau)$ in (5.11) will be a constant value 1, which results in a large error in estimating expected revenue of taxis. With our proposed graph partition algorithm, the spatiotemporal correlation between the links within a community is stronger than that in different communities. It has benefit for accurately forecasting $y_{h_i,\tau}, h_i \in H$.

The complexity of SLPA is $O(\mathcal{T}|E|)$. The complexity of GN algorithm depends on the results partitioned by SLPA. Consider the number of communities whose sizes are larger than ϵ_{size} is $N_{\epsilon_{size}}, N_{\epsilon_{size}} \leq \frac{|U|}{\epsilon_{size}}$. Suppose the maximal number of edges and vertices in $N_{\epsilon_{size}}$ large communities are N_{edge} and $N_{vertices}$, respectively. Obviously, $N_{edge} \leq |E|$ and $N_{vertices} \leq |U|$. Thus, the complexity of GN algorithm is $O(N_{\epsilon_{size}} N_{edge}^2 N_{vertices}) \in O(\frac{|E|^2|U|^2}{\epsilon_{size}})$. Consequently, the complexity of GN-SLPA is $O(\mathcal{T}|E| + \frac{|E|^2|U|^2}{\epsilon_{size}})$. As GN-SLPA is an off-line algorithm and only implemented once, it has little impact on the efficiency of taxi cruising route recommendation.

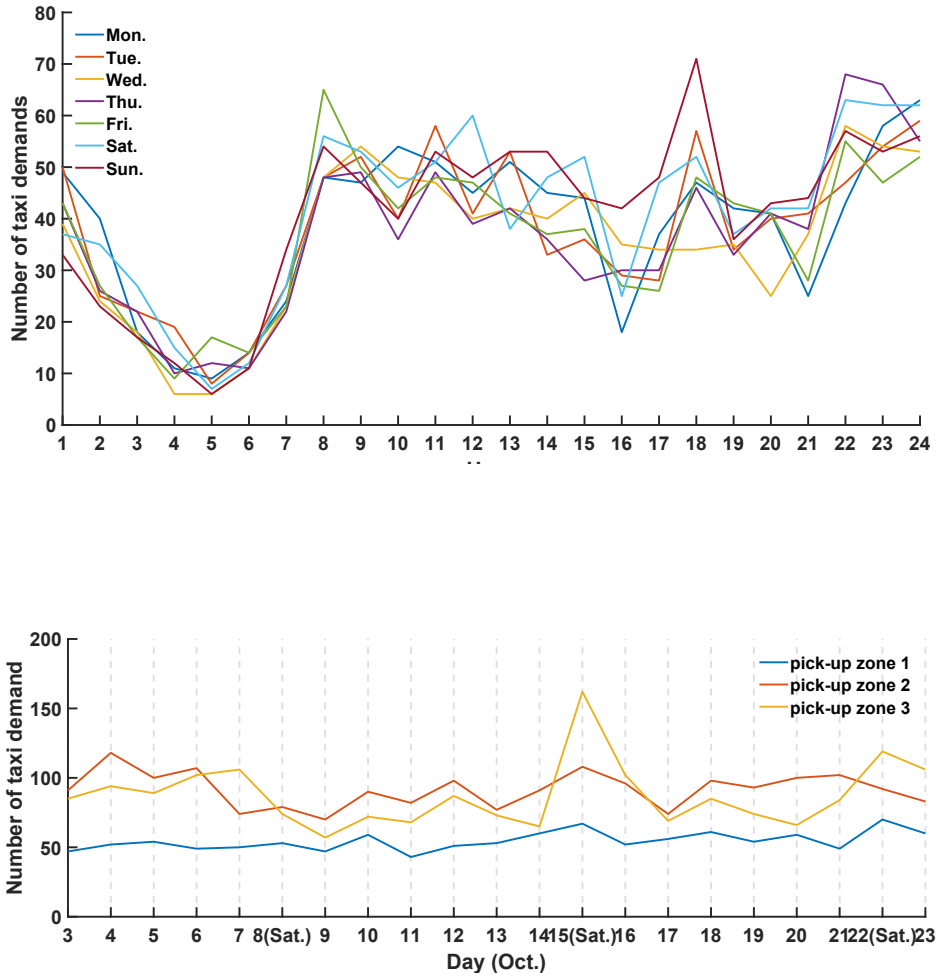
5.2.2 The Prediction of $y_{t+\tau}$

Similar with other traffic prediction problems such as traffic flow and travel speed prediction, the spatiotemporal correlation between taxi demand plays an important role in accurate forecasting and has been considered in several predictive models introduced in Section 2.3. However, compared with traffic flow or travel speed, it is more difficult to capture the spatiotemporal correlation between taxi demand in different pick-up zones. For instance, traffic flow in a link comes from its upstream neighbors. Thus, there is a strong spatiotemporal correlation between the traffic flow in neighboring links in the recent past [23]. Unlike traffic flow, taxi demand in two far-away pick-up zones could be potentially correlated in a long time period. To illustrate, consider the situation that the passengers intend to travel from A to B for working in the morning, e.g., between 7:00 am to 9:00 am. Most of them will return

to A from B after working, e.g., between 5:00 pm to 7:00 pm. There is a strong spatiotemporal correlation between taxi demand in A and B within these two rush hours. Nevertheless, in the metropolitan area, residence A and working place B are usually far away from each other. In addition, the time period between these two rush hours is large (nearly 8-12 hours). Unfortunately, such long-term spatiotemporal correlation is less considered in the existing predictors since they usually use taxi demand in the fast few hours for prediction. To solve this problem, we combine long short term memory (LSTM) technique with GNN. The graph-structured input of GNN has the benefit of representing the spatial dependency among taxi demand in different pick-up zones. Further, LSTM is capable of remembering both short and long-term temporal dependencies among taxi demand.

Apart from intricate spatiotemporal correlation, considering multiscale features of taxi demand also has the potential of improving the forecasting accuracy. To illustrate, Fig.5.3 (more details of data and partition results will be illustrated in Section 5.3) depicts the variation of taxi demand under time scales of an hour and a day, respectively. Compared Fig.5.3a with Fig.5.3b, we can observe that there is obvious difference between the patterns of taxi demand variation under different time scales. For instance, there are two peak hours (7-8 am, 5-6 pm) and two off-peak hours (4-5am, 8-9pm) in a pick-up zone during a day (Fig.5.3a). The fluctuation of taxi demand is small in the same time period of different days (Fig.5.3b). Similar results can also be observed in other pick-up zones.

Based on above analysis and observation, we design a multi-scale GCN model based on attention enhanced LSTM neural network proposed by Si et al. [84]. Before model building, we first illustrate graph structured input of our proposed model and then briefly introduce the principle of graph convolution. Lastly, we present the the proposed LSTM-GCN model.



(b) Taxi demand of three pick-up zones between 8 am to 9 am in each day of three weeks from 3rd, Oct. to 23th, Oct. 2016.

Figure 5.3 : An instance of SLPA based algorithm

Graph Structured Input

Based on the partitioned result of G, we denote the graph structure of an input as $\mathcal{G} = \{H, \mathcal{E}\}$. Each vertex $h_i \in H \subseteq U$ represents a pick-up zone based on Definition 1. There is an undirected edge $\mathcal{E}_j \in \mathcal{E}$ between $h_{i_1} \in H$ and $h_{i_2} \in H$ if there is at least one edge connecting $u_1 \in h_{i_1}$ and $u_2 \in h_{i_2}$. In the time interval $(t - \omega\tau, t - (\omega - 1)\tau]$, a signal on graph $\mathcal{G}$ is the taxi demand over the whole road network, that is, $\boldsymbol{y}_{t-\omega\tau}$.

Graph Convolution

Graph convolution heavily relies on graph Fourier operation implemented on the normalized Laplacian matrix of a undirected graph denoted by $L = I - D^{\frac{1}{2}}AD^{\frac{1}{2}}$. I is identity matrix, A is adjacent matrix of $\mathcal{G}$ and D is diagonal matrix where each diagonal element $D_{ii} = \sum_j(A_{ij})$. L can be further decomposed into $L = Q\Lambda Q$, where Q is the matrix of eigenvectors, and Λ is a diagonal matrix where each diagonal element Λ_{ii} is an eigenvalue. We define the attributes associated with each node in the graph as graph signals, denoted as a a vector y. Hence, the graph Fourier transform to y is defined as $\mathscr{F}(y) = Q^T y$. The graph convolution of graph signal y and a convolutional kernel κ (a vector, also known as filter) is defined as

$$\begin{aligned} y *_{\mathcal{G}} \kappa &= \mathscr{F}^{-1}(\mathscr{F}(y) \odot \mathscr{F}(\kappa)) \\ &= Q(Q^T y \odot Q^T \kappa). \end{aligned} \tag{5.14}$$

In 5.14, $*_{\mathcal{G}}$ is the convolution on the undirected graph $\mathcal{G}$,$\mathscr{F}^{-1}$ is the inversed Fourier transformation, defined as $\mathscr{F}^{-1}(y) = Qy$, and $\odot$ is the Hadamard product. We define a diagonal matrix Θ where each diagonal element $\Theta_{ii} = Q^T(i)\kappa'$. Particularly, $Q^T(i)$ is the i-th row of Q^T and κ' is the transposition of κ. Then, we rewrite (5.14) as

$$y *_{\mathcal{G}} \kappa = Q(Q^T y \odot Q^T \kappa) = Q\Theta Q^T y. \tag{5.15}$$

There are a lot of available kernels. More details of these kernels can be referred to [31].

Multi-scale LSTM-GCN model

As shown in Fig.5.4, there are two identical LSTM based sub-networks. The structure of an LSTM component is shown on the left side of Fig. 5.4. Similar to LSTM in RNN, there are also three gates, namely input gate, forget gate and output gate in our proposed LSTM component. Different from conventional LSTM

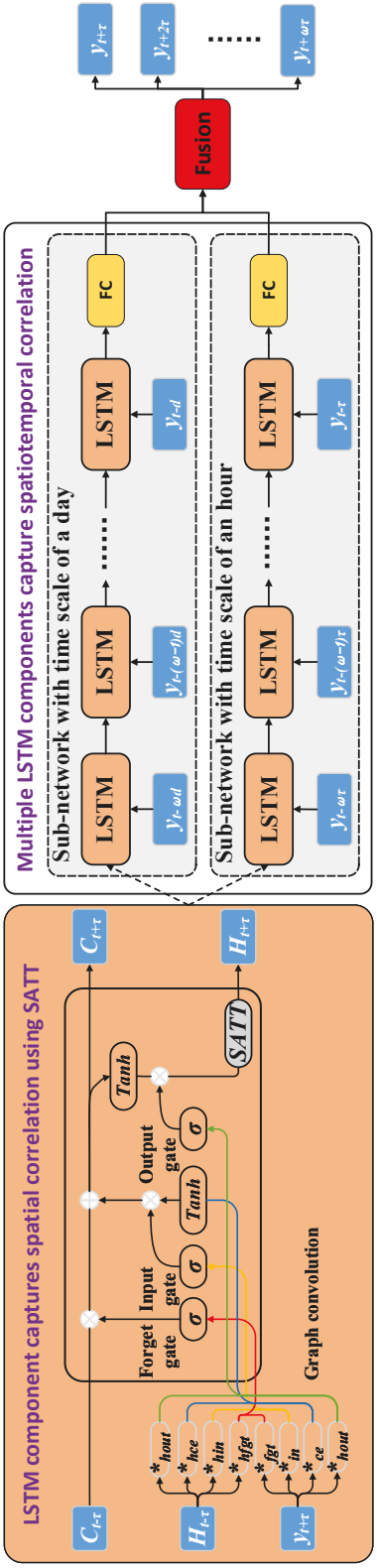


Figure 5.4 : The structure of multiscale LSTM-GCN

and RNN, in our LSTM component, the functions of these three gates are based on graph convolution which is defined as follows:

$$\begin{aligned} \boldsymbol{i}_{t+\tau} &= \sigma(\boldsymbol{y}_{t+\tau} *_{\mathcal{G}} \kappa_{in} + \boldsymbol{H}_{t-\tau} *_{\mathcal{G}} \kappa_{hin} + \boldsymbol{b}_{in}) \\ \boldsymbol{f}_{t+\tau} &= \sigma(\boldsymbol{y}_{t+\tau} *_{\mathcal{G}} \kappa_{fgt} + \\ &\quad \boldsymbol{H}_{t-\tau} *_{\mathcal{G}} \kappa_{hfgt} + \boldsymbol{b}_{fgt}) \\ \boldsymbol{o}_{\tau} &= \sigma(\boldsymbol{y}_{t+\tau} *_{\mathcal{G}} \kappa_{out} + \boldsymbol{H}_{t-\tau} *_{\mathcal{G}} \kappa_{hout} + \boldsymbol{b}_{out}), \end{aligned} \tag{5.16}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function and $\boldsymbol{b}$s are the bias. With (5.15) and Chebyshev polynomial kernel $\Theta = \sum_{j=0}^{\mathcal{K}} \alpha_j \Lambda^{\mathcal{K}}$, $y *_{\mathcal{G}} \kappa$ in (5.16) can be expressed as

$$\begin{aligned} \boldsymbol{y} *_{\mathcal{G}} \kappa &= Q(\sum_{j=0}^{\mathcal{K}} \alpha_j \Lambda^{\mathcal{K}}) Q^T \boldsymbol{y} \\ &= \sum_{j=0}^{\mathcal{K}} \alpha_j (Q \Lambda^{\mathcal{K}} Q) \boldsymbol{y} \\ &= \sum_{j=0}^{\mathcal{K}} \alpha_j L^{\mathcal{K}} \boldsymbol{y} \end{aligned} \tag{5.17}$$

From (5.17), we can observe that graph convolution based on Chebyshev polynomial kernel is the function of L.

The hidden state ($\boldsymbol{H}_{\tau}$) in our proposed LSTM component is estimated as:

$$\begin{aligned} \boldsymbol{u}_{t+\tau} &= \tanh(\boldsymbol{y}_{t+\tau} *_{\mathcal{G}} \kappa_{ce} + \boldsymbol{H}_{t-\tau} *_{\mathcal{G}} \kappa_{hce} + \boldsymbol{b}_{ce}) \\ \boldsymbol{C}_{t+\tau} &= \boldsymbol{f}_{t+\tau} \odot \boldsymbol{C}_{t-\tau} + \boldsymbol{i}_{t+\tau} \odot \boldsymbol{u}_{t+\tau} \\ \hat{\boldsymbol{H}}_{t+\tau} &= \boldsymbol{o}_{t+\tau} \odot \tanh(\boldsymbol{C}_{t+\tau}) \\ \boldsymbol{H}_{t+\tau} &= f_{att}(\hat{\boldsymbol{H}}_{t+\tau}), \end{aligned} \tag{5.18}$$

where $\boldsymbol{u}_{t+\tau}$ is modulated input, $\boldsymbol{C}_{t+\tau}$ is the cell state and $\hat{\boldsymbol{H}}_{t+\tau}$ is an intermediate hidden state. $f_{att}(\cdot)$ is based on the the spatial attention mechanism proposed by Feng et al. [27], and formulated as:

$$f_{att}(\hat{\boldsymbol{H}}_{t+\tau}) = \gamma \hat{\boldsymbol{H}}_{t+\tau}, \tag{5.19}$$

where γ is a $|H| \times |H|$ matrix. Each element in γ, γ_{ij} denotes the similarity between $\hat{\boldsymbol{H}}_{t+\tau,i}$ and $\hat{\boldsymbol{H}}_{t+\tau,j}$, and calculated as:

$$\gamma_{ij} = \gamma_{ji} = \frac{\exp(\hat{\boldsymbol{H}}_{t+\tau,i} W_{ij} \hat{\boldsymbol{H}}_{t+\tau,j})}{\sum_{k=1}^{|H|} \exp(\hat{\boldsymbol{H}}_{t+\tau,i} W_{ik} \hat{\boldsymbol{H}}_{t+\tau,k})}, \tag{5.20}$$

where W_{ij} is the (i,j)-th element in the weight matrix W. The objective of applying the spatial attention scheme is to measure the spatial correlation between taxi demand in all pick-up zones. Consider there are $\omega = \lceil 12 \ hours/\tau \rceil$ time intervals during past 12 hours (e.g., $\omega = 12$ if a time interval is an hour). Then we have ω LSTM components in the first sub-network. It aims at capturing long-term spatiotemporal dependency of taxi demand among different pick-up zones. Lastly, we use Rectified Linear Unit (ReLU) active function in the fully-connected layer (FC).

The second sub-network is similar with the first one except the time scale is a day. The input of each LSTM components is $y_{t-d\tau}$ that means the taxi demand in time interval $(t-\tau, t]$ of yesterday. For the taxi demand in time interval $(t-\tau, t]$ of the day before the yesterday is $y_{t-2d\tau}$. We will use LSTM based network to capture the spatial-temporal correlation crossing past one week ($\omega = 7$). Finally, we aggregate the information from two sub-networks as follows:

$$\boldsymbol{y}_{t+\tau} = W_{hr} \boldsymbol{y}_{hr,t+\tau} + W_{day} \boldsymbol{y}_{day,t+\tau} \tag{5.21}$$

W_{hr} and W_{day} are the learning parameters. $\boldsymbol{y}_{hr,t+\tau}$ and $\boldsymbol{y}_{day,t+\tau}$ are the predicted results of two sub-networks. All the parameters in the LSTM-GCN is learned with gradient descent method.

5.2.3 The Estimation of $\mathbf{P}(Z|H, t+\tau)$

In this paper, we estimate $\forall p(z_j|h_i, t+\tau) \in \mathbf{P}(Z|H, t+\tau)$ with a statistical method that is similar to the approaches applied in many studies like [125, 127]. Assuming we have a series of historical time intervals in which the taxi demand has a close temporal correlation with the taxi demand in $t+\tau$. For simplicity, in

this paper, we define Φ by $\Phi = \cup y_{t+d\tau}, d = [-1, -7]$ with the significance of the set of taxi demand in the same time interval during past one week. Further, we use $\mathcal{N}_\phi(h_i \to z_j)$ to denote the number of taxis traveling from h_i to z_j. After that, $p(z_j|h_i, t+\tau)$ is estimated as follows:

$$p(z_j|h_i, t+\tau) = \frac{\sum_{\phi\in\Phi} \mathcal{N}_\phi(h_i \to z_j)}{\sum_{z_j\in Z}\sum_{\phi\in\Phi} \mathcal{N}_\phi(h_i \to z_j)}, h_i \in H \tag{5.22}$$

5.2.4 Distributed Algorithm

To simplify the procedure of problem solving, given the n-th vacant taxi, $n \in N_t$, we calculate $R(x_{nm} = 1), \forall m \in M_{t+\tau}$. After that, we filter the taxi demand with which $R(x_{nm} = 1) < \epsilon_{rev}$. In this case, we use $M^n_{t+\tau}$ to denote available taxi demand for the n-th taxi after filtering. Thus, for any $m \in M^n_{t+\tau}$, $R(x_{nm} = 1) \geqslant \epsilon_{rev}$ and constraint (5.4) is satisfied. Further, we relax (5.5) by $0 \leqslant x_{nm} \leqslant 1$ in order to transform $\mathbf{P}^{'}$ to a continuous convex program. In practice, the relaxation has no impact on the solution. It will be shown later.

With above operations, we rewrite $\mathbf{P}^{'}_{\mathbf{x}}$ as follows:

$$\mathbf{P}^{'}: \quad \min \ w_1 f_1(\boldsymbol{x}_{t+\tau}) - w_2 f_2(\boldsymbol{x}_{t+\tau}) \tag{5.23}$$

$$s.t. \quad \sum_{m\in M^n_{t+\tau}} x_{nm} \leqslant 1, \forall n \in N_t \tag{5.24}$$

$$\sum_{n\in N_t} x_{nm} \leqslant 1, \forall m \in M^n_{t+\tau} \tag{5.25}$$

Meanwhile, we introduce $M^n_{t+\tau}$ Lagrange multipliers $\mu = \{\mu_m | \mu_m \geqslant 0, \forall m \in M^n_{t+\tau}\}$. By taking μ and constraint (5.3) inside the objective, we get the Lagrangian of $\mathbf{P}^{'}$, $\mathcal{L}(\boldsymbol{x}_{t+\tau}, \mu)$, expressed by (5.27). Based on $\mathcal{L}(\boldsymbol{x}_{t+\tau}, \mu)$, we formulate the dual problem of $\mathbf{P}^{'}$, $\mathcal{D}(\mathbf{P}^{'})$ as:

$$\begin{aligned} \mathcal{D}(\mathbf{P}^{'}): \quad & \max_{\mu} \ \min_{\boldsymbol{x}_{t+\tau}} \mathcal{L}(\boldsymbol{x}_{t+\tau}, \mu) \\ s.t. \quad & (5.24), (5.25) \end{aligned} \tag{5.27}$$

Algorithm 5.1: Distributed dual ascent based algorithm

Client

Initialization: $M_{t+\tau}^n \leftarrow M_{t+\tau}$

for $\forall m \in M_{t+\tau}$ **do**

  Calculate $R(x_{nm}=1)$ based on (5.11) and (5.9)

  if $R(x_{nm}=1) < \epsilon_{rev}$ **then**

    $M_{t+\tau}^n = M_{t+\tau}^n - m$

  end

end

receive $\mu^{(k)}$ from terminal

if ***MSG_TERMINATE = FALSE*** **then**

  $x_{nm^*}^{(k)} = 1$, $m^* = \underset{m \in M_{t+\tau}^n}{\operatorname{argmax}}\{Q_{x_{nm}=1}\}$

  send $x_{nm^*}^{(k)} = 1$ to terminal

else

  $x_{nm^*}^{(k-1)}$ is the solution

end

Terminal

Initialization: $\mu^{(0)}$, $k=0$, **MSG_TERMINATE = FALSE**

Send $\mu^{(k)}$ to client

$k \leftarrow k+1$

Receive $\boldsymbol{x}_{t+\tau}^{(k)}$ from client

Update $\mu_m^{(k)}, \forall m \in M_{t+\tau}^n$ with (5.32)

if $\mu^{(k)}$ *converges* **then**

  MSG_TERMINATE=TRUE

end

send **MSG_TERMINATE** to client

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_{t+\tau}, \mu) &= w_1 f_1(\boldsymbol{x}_{t+\tau}) - w_2 f_2(\boldsymbol{x}_{t+\tau}) + \sum_{m \in M^n_{t+\tau}} \mu_m (\sum_{n \in N_t} x_{nm} - 1) \\
&= w_1 (N_t - \sum_{n \in N_t} \sum_{m \in M^n_{t+\tau}} x_{nm}) - w_2 \sum_{n \in N_t} \sum_{m \in M^n_{t+\tau}} R(x_{nm}) \\
&\quad + \sum_{m \in M^n_{t+\tau}} \mu_m \sum_{n \in N_t} x_{nm} - \sum_{m \in M^n_{t+\tau}} \mu_m \\
&= w_1 N_t - \left(\sum_{n \in N_t} \sum_{m \in M^n_{t+\tau}} (w_1 - \mu_m) x_{nm} + w_2 R(x_{nm}) \right) - \sum_{m \in M^n_{t+\tau}} \mu_m .
\end{aligned} \tag{5.26}$$

Suppose the optimal solution of $\mathbf{P}'$ and $\mathcal{D}(\mathbf{P}')$ are $\boldsymbol{x}^*_{t+\tau}$ and $\bar{\boldsymbol{x}}^*_{t+\tau}$, respectively. As $\mathbf{P}'$ is a convex problem, strong duality is hold. Thus, we have $\boldsymbol{x}^*_{t+\tau} = \bar{\boldsymbol{x}}^*_{t+\tau}$. To find $\bar{\boldsymbol{x}}^*_{t+\tau}$, we design a distributed algorithm (Algorithm 5.1) based on the dual ascent method, which is implemented at both of terminal (server) and client (applications in the smart phone or some other devices equipped at each taxi) of the route recommendation system. The core idea of Algorithm 5.1 is as follows. $M^n_{t+\tau}$ is estimated from line 1 to 6 at each client. In the k-th ($k \geqslant 1$) iteration, $x_{nm} \in \boldsymbol{x}^{(k)}_{t+\tau}$ is updated at the n-th client with $\mu^{(k-1)}$ sent from terminal (line 7-10 at client). Particularly, when $k = 1$, $\boldsymbol{x}^{(1)}_{t+\tau}$ is updated with the initial values of μ, denoted by $\mu^{(0)}$. After receiving termination message (**MSG_TERMINATE** = **TRUE**), the optimal solution for the n-th taxi is obtained by line 11-13 at client. With $\boldsymbol{x}^{(k)}_{t+\tau}$, $\mu^{(k)}$ is updated at the terminal (line 3 and 4). The algorithm terminates when $\mu^{(k)}$ converges, that is, $\mu^{(k)} = \mu^{(k-1)}$ (line 5-7). Lastly, **MSG_TERMINATE** is sent to each client. In the following, we illustrate the process with respect to the update of $\boldsymbol{x}^{(k)}_{t+\tau}$ and $\mu^{(k)}$. First, we define a Lagrange dual function $g(\mu^{(k)})$ by

$$\begin{aligned}
g(\mu^{(k)}) &= \min_{\boldsymbol{x}_{t+\tau}} \mathcal{L}(\boldsymbol{x}_{t+\tau}, \mu^{(k)}) \\
&\quad s.t. \quad (5.24), (5.25)
\end{aligned} \tag{5.28}$$

and separate $g(\mu^{(k)})$ in terms of each taxi as follows:

$$g(\mu^{(k)}) = -\sum_{n \in N_t} g_n(\mu^{(k)}) + \min_{\boldsymbol{x}_{t+\tau}}(w_1 N_t - \sum_{m \in M^n_{t+\tau}} \mu_m^{(k)}), \tag{5.29}$$

where $g_n(\mu^{(k)})$ is in the form:

$$\begin{aligned} g_n(\mu^{(k)}) \quad &= \quad \max_{x_{nm}} \textstyle\sum_{m \in M^n_{t+\tau}} (w_1 - \mu_m^{(k)}) x_{nm} + w_2 R(x_{nm}) \\ s.t. \quad & \qquad\qquad (5.24), (5.25) \end{aligned} \tag{5.30}$$

Obviously, for the fixed $\mu^{(k)}$, the optimal solution of $g(\mu^{(k)})$ is obtained by solving $g_n(\mu^{(k)})$ for all $n \in N_t$ since $\min_{\boldsymbol{x}_{t+\tau}}(w_1 N_t - \sum_{m \in M^n_{t+\tau}} \mu_m^{(k)})$ in (5.29) is constant. We define $Q^{(k)}_{x_{nm}=1} = w_1 - \mu_m^{(k)} + w_2 R(x_{nm})$, then the optimal solution for $g_n(\mu^{(k)})$ with the fixed $\mu^{(k)}$ is $x_{nm^*} = 1$ where m^* is

$$m^* = \underset{m \in M^n_{t+\tau}}{\operatorname{argmax}} \{Q^{(k)}_{x_{nm}=1}\}, \forall n \in N_t \tag{5.31}$$

Note that there may be several ms with which $Q^{(k)}_{x_{nm}=1}$ is the maximum of all. In this case, we randomly select one as m^*.

After finding $x^{(k)}_{nm^*}$s for all $n \in N_t$ by the fixed $\mu_m^{(k)}$s, we get $\boldsymbol{x}^{(k)}_{t+\tau}$ and update $\mu_m^{(k+1)}$ for each $m \in M^n_{t+\tau}$ by the following gradient descent method [11]:

$$\mu_m^{(k+1)} = \left[\mu_m^{(k)} - \beta(k) \left(1 - \sum_{n \in N_t} x_{nm} \right) \right]^+ . \tag{5.32}$$

As $\mu_m \geqslant 0, \forall m \in M^n_{t+\tau}$, the operator $[\cdot]^+$ in (5.32) indicates the maximum of $\mu_m^{(k)} - \beta(k) \left(1 - \sum_{n \in N_t} x_{nm}\right)$ and 0. Furthermore, We define $\beta(k) = \frac{0.5}{k}$ with the purpose of controlling the convergence speed of μ.

5.3 Results

5.3.1 Experimental Setting

As shown in Fig.5.5, our study site is based on the citywide road network in Xi'an, China. With SUMO (Simulation of Urban MObility) and OpenStreetMap,

we extract 30,549 links, including the primary, secondary, and side roads, which are classified by Xi'an planning bureau. There are over 11,000 taxicabs anonymously reporting GPS trajectories to the server with an average sampling frequency of 30 seconds. We yield over 2.7e+09 raw data records crossing 90 days from 1st, Sep. 2016 to 29th, Nov. 2016. Each data record had the information including the time stamp when the data was reported, the geographical coordinates (longitude and latitude), the instantaneous travel speed and travel state taking values from**{stop, cruising, occupied}**

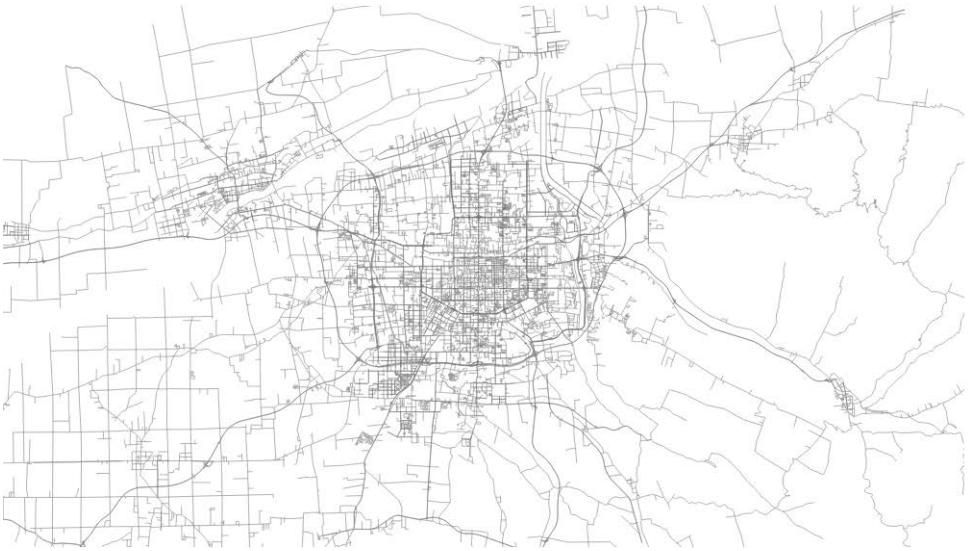


Figure 5.5 : The study site

We divide the day into 24 equal time intervals, denoted by $\{\tau_i | i = 1, 2, ..., 24\}$ where $\forall \tau_i$ represents an hour, e.g., 8:00am-9:00am. Also, we implement map matching since the GPS trajectories of collected data usually do not fall into the links correctly.

5.3.2 Graph Partition

From Fig.5.6, we can observe that the number of communities partitioned by SLPA algorithm will converge to around 1430 after 1200 iterations. Therefore, we

set the parameter $\mathcal{T}$ by 1200 and obtain 1468 communities. With $\epsilon_{size} = 120$, we further divide large community (e.g., a community with 1690 links) into small ones. After the implementation of GN algorithm, we have 1945 communities.

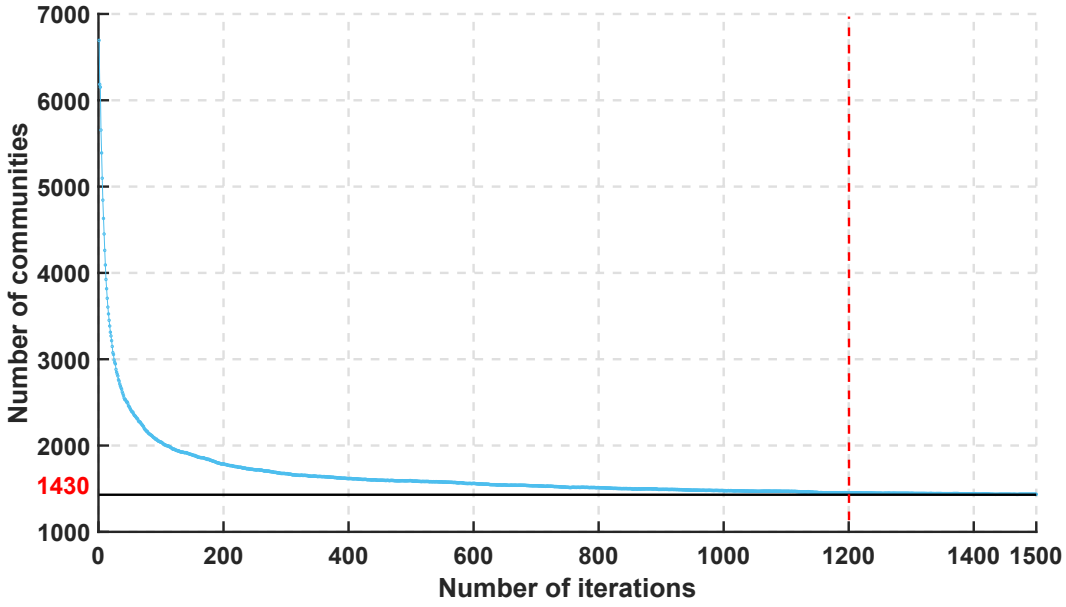


Figure 5.6 : The number of communities based on SLPA with different iterations

The statistic of different sizes of communities is presented in Fig 5.7. Compared with the partitioned results with only SLPA, GN algorithm is capable of reducing the size of communities.

5.3.3 Taxi Demand Prediction

The setting of the proposed LSTM-GCN is as follows. The number of the terms of Chebyshev polynomial kernel κ is 3. The larger value can improve the prediction accuracy but consumes more computational resource. As τ is one hour, we have 12 LSTM components in the first sub-network. 7 LSTM components are set in the second sub-network. After model training, we make single step prediction, that is, the taxi demand in next one hour. The batch size is 64 and learning rate is 0.0001. We use the data of 75 days to train the model and the data of 15 days to validate

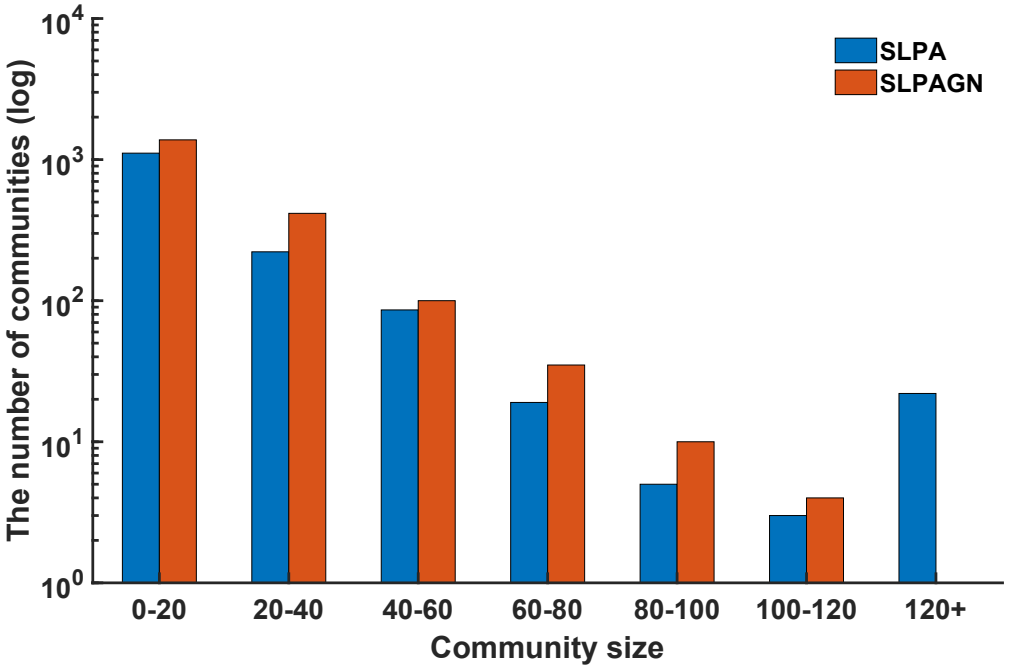


Figure 5.7 : The number of communities based on SLPA and GN-SLPA

model. We compare our predictor with the following baselines:

- Historical Average (HA) model. In this paper, the predicted results is the average value of taxi demand in the historical ten time intervals.
- Auto-Regressive Integrated Moving Average (ARIMA) model. It is a time series based model and has been widely used for traffic prediction like traffic flow.
- Vector Auto-Regressive (VAR) model. It is another time series forecasting model.
- STGCN model. It is used for traffic flow prediction by considering the spatiotemporal correlation between traffic.
- Single LSTM-GCN (SLSTM-GCN). Compared with our proposed LSTM-GCN, SLSTM-GCN only has one network with only considering the time scale of an hour. Besides, the number of LSTM components is 3.

To measure the performance of these predictors, we use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

Table 5.3 : The performance of predicted taxi demand

Model	RMSE	MAE
HA	55.62	39.74
ARIMA	48.21	35.16
VAR	51.94	37.67
STGCN	35.64	28.59
SLSTM-GCN	34.89	28.35
LSTM-GCN	33.58	26.92

In Table 5.3, we can observe that the best results are obtained from our method. There is no doubt that the worst results are obtained with HA method. The gap between STGCN and our proposed method is not as much as the ones between our method and others. This can be explained by the fact that STGCN also considers spatiotemporal correlation between taxi demand. However, the performance of STGCN is a little less than our proposed method since they do not consider the long-term spatiotemporal dependencies. This can also explain the fact that the predicted results of SLSTM-GCN and STGCN are similar. Compared with SLSTM-GCN and LSTM-GCN, we can observe that considering multiple time scale has the benefit of improving the prediction accuracy.

5.3.4 Simulation

As the performance of a scheme for the taxi cruising route recommendation can be observed only under the case that it is implemented in real life, it is difficult to carry out experiments. In this case, we implement simulation based on the road

network of Xi'an with the aid of SUMO and OpenStreetMap. Moreover, we assume taxi demand in each pick-up location follows the one-dimensional space-time Poisson process distribution where the parameter $\lambda = 6/hour/zone$. We use the method in Section 5.2.3 to estimate the distribution of taxi destinations. We compare our proposed scheme with the following three baselines:

- Random walk. The vacant taxis pick up the passengers randomly and cruise along the path with shortest length.
- Optimal policy. The scheme is based on the MDP which consider the long-term profit.
- Single objective based distributed scheme. It is based on our proposed distributed scheme where only objective $f_2(\boldsymbol{x}_{t+\tau})$ is considered.

We measure the performance of different schemes from the perspectives of average revenue and occupancy rate. The occupancy rate is defined as the ratio of the number of occupied taxis and the number of all the taxis. Further, in Table 5.4, we show the performance of the scheme in three time intervals of a day.

From Table 5.4, we can observe that the average revenue of our distributed scheme and optimal policy is similar. However, our method can effectively reduce the number vacant taxis. Besides, the occupancy rate of our proposed scheme is much larger than the single objective based distributed scheme. For instance, our proposed scheme reduce almost 960 vacant taxis compared with single objective based scheme. It further validates our intuition that problem formulation with single objective easily leads to more vacant taxis.

Table 5.4 : The performance between our proposed distributed scheme and other counterparts

Scheme	Average revenue				Occupancy rate			
	7:00-8:00	11:00-12:00	17:00-18:00	A day	7:00-8:00	11:00-12:00	17:00-18:00	A day
Random walk	48.25	34.12	51.33	38.57	0.40	0.42	0.45	0.41
Single objective	62.37	43.46	59.82	45.18	0.51	0.49	0.50	0.47
Optimal policy	59.49	41.17	60.88	46.32	0.53	0.50	0.51	0.49
Distributed scheme	60.18	42.95	58.41	44.91	0.58	0.53	0.54	0.55

5.4 Summary

In this chapter, we developed a distributed scheme for taxi cruising route recommendation. To this end, an SLPA and GN based algorithm was designed to partition the urban road network into a series of communities, namely pick-up zones. By modeling the partitioned results as an undirected graph, an LSTM-GCN model was proposed to forecast taxi demand in each pick-zone. Based on the predicted taxi demand and taxi destinations, we develop a Lagrange dual decomposition-based method to obtain the solution to the bi-objective optimal problem for taxi recommendation. Experimental results using real trajectory data collected from taxis in Xi'an, China show that LSTM-GCN model has a better performance than its counterparts. The simulation results show that the solution obtained with our proposed scheme is better than its counterparts, such as random work, and has fewer vacant taxis compared with the single-objective based problem formulation.

Chapter 6

Conclusion

This thesis had presented works on modeling, analysis and application of big traffic data in ITS, including developing a novel short-term traffic flow predictor, a novel estimator for link travel time estimation and effective scheme for taxi cruising route recommendation. In the following text, the key results and findings of this thesis are summarised.

In chapter 3, "Unified Spatio-temporal Model for Short-term Traffic Flow Prediction," we developed a unified spatiotemporal model based on STARIMA. Before model building, we analyzed the potential factors affecting the time-varying spatiotemporal correlation between traffic on different roads. More precisely, these factors included road topology, time-varying travel speed, and trip distribution. After that, we captured such time-varying spatiotemporal correlation with the aid of a series of parameters by considering incomplete traffic data in the range of urban road networks. Finally, we integrated these parameters with the traditional STARIMA model and obtain our proposed unified STARIMA model. Thus, the parameters of the developed predictor had physically intuitive meanings, which made the model readily amendable to suit changing road topology and traffic conditions. Experiments using real traffic data showed that the proposed approach had superior accuracy compared with their counterparts, namely STARIMA and BPNN, meanwhile had a much reduced computational complexity.

In chapter 4, "Estimation of Link Travel Time Distribution With Limited Traffic Detectors," we proposed a KDE based model for link TTD estimation. Different

from most existing methods where parametric models were used for modeling link TTD, our proposed model was based on a non-parametric model, which was more feasible for link TTD modeling under the different road conditions (free flow or congestion). To accurately and efficiently estimate the parameters in the proposed model, we proposed a set of strategies, including C-shortest path algorithm, K-means based algorithm, the EM algorithm, Q-opt algorithm and X-means based algorithm. Experimental results using real data collected by the taxicabs in Xi'an, China showed that the TTDs estimated using our proposed model were in excellent agreement with empirical distribution, but only with limited traffic detectors configured at partial intersections.

In chapter 5, "Graph Neural Network And Distributed Lagrange Dual Decomposition Based Method For Taxi Route Recommendation," we developed a MIP scheme to model taxi cruising route recommendation. With data analysis, we found that the solutions provided by most of the existing taxi recommendation systems easily falled into the local optimum due to the single objective based problem formulation. To solve this problem, in our proposed scheme, we formulated bi-objectives including minimizing the number of vacant taxis to gain individual revenue and maximizing global revenue by considering collaboration and competition between taxis. To get the solution, we first estimated taxi demands in different areas partitioned using a joint SLPAGN algorithm by considering road topology and geographical distribution of POIs. After that, we proposed a LSTM-GCN based method integrating multi-scale features and spatiotemporal correlation in taxi demands, which had the potential of improving the forecasting accuracy. Finally, we employed a distributed algorithm based on Lagrange dual decomposition to get the optimal solution. Both experimental results using real data and simulation results showed that the proposed scheme had a better performance than their counterparts from the perspectives of taxi demand prediction and taxi cruising route recommendation.

In addition to the encouraging results and findings summarised above, there are still some research problems to be investigated in the future. First, although our proposed unified STARIMA based model has a good forecasting performance, it is unable to capture the non-linear trend of traffic data. This problem can be overcome by deep learning techniques, which have attracted more and more attention. However, as analyzed in this thesis, existing deep learning based methods are black-box ones in which the parameters lack physical illustration, thereby, it is part of our future work to apply our proposed strategy into deep learning techniques so that the accuracy of traffic prediction may be further improved. Second, despite using limited traffic detectors to estimate link TTD, the strategy proposed in our method for traffic detector deployment is not the optimal one if we consider the objectives of minimizing the economic cost and optimizing the estimation accuracy. There is a conflict between the two aforementioned objectives. For instance, the more detectors we use, the more data we collect. Therefore, good estimated results will be obtained based on the data-driven method. However, there will be a higher economic cost for traffic detector deployment. Thus, it is part of our future work to explore a strategy to guarantee the trade-off between estimation accuracy and traffic detector placement. Furthermore, the proposed scheme of taxi recommendation focuses on taxis only. Currently, not only taxis but also private vehicles have the same functions as taxis due to the utilization of e-hailings applications like Uber and Didi. There should be competition and collaboration between private vehicles and taxis. However, there is less research concerning the policy of regulating the services of private vehicles and taxis, causing a lot of social problems such as the conflict between the revenue obtained by the drivers of private vehicles and taxis. Moreover, it may produce heavier traffic problems and fuel consumption due to the excessive vehicles for passengers, especially in off-peak hours. As a result, it is necessary to model the potential connection between private vehicles and taxis, and

then develop a policy to better passenger allocation, so that a win-win destination can be achieved among private vehicles and taxis.

Appendices

Appendix A

M-step in EM Algorithm in Section 4.2.3

We use $\mathbb{T}_{r_j}$to denote the E2E measurements collected on $r_j \subseteq \mathbb{R}$. Then $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$ can be rewritten as

$$\begin{aligned}\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}}) &= \sum_{r_j \subseteq \mathbb{R}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{k=1}^{d_{r_j}} \ln \frac{1}{h_{e_k}} \\ &+ \sum_{r_j \subseteq \mathbb{R}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \ln \sum_{z=1}^{\mathcal{Z}_{r_j}} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2). \end{aligned} \tag{A.1}$$

Further, we use $\mathbb{R}_{e_k}$ to represent the set of paths which cover e_k. Then we take the derivatives of $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$ with respect to $\mu_{e_k,i} \in \mu_{e_k}$in $\Theta_{e_k} \subseteq \mathbf{\Theta}_{\mathbb{R}}$. As $\mu_{r,z} = \sum_{k=1}^{d_r} u_{e_k,i}, \forall i \in n_{e_k}$, $\frac{\partial \mu_{r_j,z}}{\partial \mu_{e_k,i}} = 1$.Thus, $\frac{\partial \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\partial \mu_{e_k,i}}$ is formulated as (A.2) where $\gamma_{t_{r_j}}(y_z)$ is the responsibility.

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\partial \mu_{e_k,i}} &= \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \left(\frac{\partial \ln \sum_{z=1}^{\mathcal{Z}_{r_j}(\mu_{e_k,i})} \mathcal{N}(t_{r_j}|\mu_{r_j,z}, h_{r_j}^2)}{\partial \mu_{r_j,z}} \cdot \frac{\partial \mu_{r_j,z}}{\partial \mu_{e_k,i}} \right) \\ &= \sum_{r_j \subseteq \mathbb{R}_{e_k}} \frac{\sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z=1}^{\mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}(y_z)(\mu_{r_j,z} - t_{r_j})}{2h_{r_j}^2}, \end{aligned} \tag{A.2}$$

Setting $\frac{\partial \mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})}{\partial \mu_{e_k,i}}$ to zero, we find that it is also difficult to calculate $\mu_{e_k,i}$ since h_{r_j}s for $\forall r_j \subseteq \mathbb{R}_{e_k}$ are different. To simplify the calculation, we assume that h_{e_k}s on $\forall e_k \in E$ are the same (*Assumption 1*). With N_{r_j} defined in (4.20), we obtain $u_{e_k,i}$

as follows:

$$u_{e_k,i} = \frac{1}{N_{r_j}} \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}(y_z) t_{r_j}. \tag{A.3}$$

Similarly, setting the derivative of $\mathcal{L}(\mathbb{T}|\mathbf{\Theta}_{\mathbb{R}})$ with respect to $h_{e_k}$to zero, we have

$$h^2_{e_k} = \frac{1}{N_{r_j}} \sum_{r_j \subseteq \mathbb{R}_{e_k}} \sum_{t_{r_j} \in \mathbb{T}_{r_j}} \sum_{z \in \mathcal{Z}_{r_j}(\mu_{e_k,i})} \gamma_{t_{r_j}}(y_z)(t_{r_j} - u_{e_k,i})^2. \tag{A.4}$$

In the q-th iteration of M-step, we update $u^{(q)}_{e_k,i}$ and $(h^{(q)}_{e_k})^2$ with (A.3) and (A.4) using the responsibilities evaluated with the parameters $\mathbf{\Theta}^{(q-1)}_{\mathbb{R}}$.

Appendix B

Estimation of SMS with TMS in Section 3.2.1 and 4.3.2

We denote the instantaneous speed of a vehicle i traveling on a link by $v_{i,ins}$. Further, we use v_{tms} and t_{sms} to denote time mean speed (TMS) and space mean speed (SMS) of the vehicles traveling on the same link. Due to the limited number and low sampling frequency of GPS trajectories, in this paper, we regard t_{sms} as an approximate value of real SMS for the i-th vehicle. More precisely, based on [24], we have the relationship between TMS and SMS as follows:

$$v_{tms} = v_{sms} + \frac{\sigma^2}{v_{sms}}, \tag{B.1}$$

where $\sigma^2 = E((v_{i,ins} - v_{sms})^2)$ and $E(v_{i,ins}) = v_{tms}$. Then, the solution to (B.1), v_{sms}, can be obtained as follows :

$$v_{sms} = \frac{3v_{tms} + \sqrt{9v_{tms}^2 - 8E(v_{i,ins}^2)}}{4} \tag{B.2}$$

Jiang et al. [36] assumed a quadratic relationship between $E(v_{i,ins}^2)$ and $E(v_{i,ins})$: $E[v_{i,ins}^2] = aE(v_{i,ins})^2 + bE(v_{i,ins}) + c$ where the parameters $\{a, b, c\}$ were estimated using 9304 samples as $\{a, b, c\} = \{1.22, -15.21, 207.95\}$. We estimate $E(v_{i,ins})$ by

$$E(v_{i,ins}) = \sum_{i=1}^{n} v_{i,ins}/n, \tag{B.3}$$

where n is the number of GPS trajectories collected on the link. Substituting (B.3) into (B.2), we have v_{sms}.

Bibliography

[1] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[2] J. Ahn, E. Ko, and E. Y. Kim, "Highway traffic flow prediction using support vector regression and bayesian classifier," in *2016 International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2016, pp. 239–244.

[3] H. Al-Deek and E. B. Emam, "New methodology for estimating reliability in transportation networks with degraded link capacities," *Journal of intelligent transportation systems*, vol. 10, no. 3, pp. 117–129, 2006.

[4] B. Barbagli, G. Manes, R. Facchini, and A. Manes, "Acoustic sensor network for vehicle traffic monitoring," in *Proceedings of the 1st International Conference on Advances in Vehicular Systems, Technologies and Applications*, 2012, pp. 24–29.

[5] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, "Network state estimation and prediction for real-time traffic management," *Networks and Spatial Economics*, vol. 1, no. 3-4, pp. 293–318, 2001.

[6] A. Bhaskar, M. Qu, and E. Chung, "Bluetooth vehicle trajectory by fusing bluetooth and loops: Motorway travel time statistics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 113–122, 2015.

[7] C. Biernacki, G. Celeux, and G. Govaert, "Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture

models," *Computational Statistics & Data Analysis*, vol. 41, no. 3-4, pp. 561–575, 2003.

[8] D. Billings and J.-S. Yang, "Application of the arima models to urban roadway travel time prediction-a case study," in *2006 IEEE Int. Conf. Systems, Man and Cybernetics*, vol. 3, pp. 2529–2534.

[9] R. P. Biuk-Aghai, W. T. Kou, and S. Fong, "Big data analytics for transportation: Problems and prospects for its application in china," in *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2016, pp. 173–178.

[10] H. Boostanimehr and V. K. Bhargava, "Unified and distributed qos-driven cell association algorithms in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1650–1662, 2015.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] N. Caceres, J. Wideberg, and F. G. Benitez, "Review of traffic data estimations extracted from cellular networks," *IET Intelligent Transport Systems*, vol. 2, no. 3, pp. 179–192, 2008.

[13] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Transactions on intelligent transportation systems*, vol. 11, no. 2, pp. 485–497, 2010.

[14] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[15] T. Cheng, J. Wang, J. Haworth, B. Heydecker, and A. Chow, "A dynamic

spatial weight matrix and localized space–time autoregressive integrated moving average for network modeling," *Geographical Analysis*, vol. 46, no. 1, pp. 75–97, 2014.

[16] S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.-W. Tan, and P. Varaiya, "Traffic measurement and vehicle classification with single magnetic sensor," *Transportation Research Record*, vol. 1917, no. 1, pp. 173–181, 2005.

[17] S. Y. Cheung, S. C. Ergen, and P. Varaiya, "Traffic surveillance with wireless magnetic sensors," in *Proceedings of the 12th ITS world congress*, vol. 1917, 2005, p. 173181.

[18] G. Comert and A. Bezuglov, "An online change-point-based model for traffic parameter prediction," *IEEE Trans. Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1360–1369, 2013.

[19] N. Davis, G. Raina, and K. Jagannathan, "A multi-level clustering approach for forecasting taxi travel demand," pp. 223–228, 2016.

[20] A. De Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio, "Artificial neural networks applied to taxi destination prediction," *arXiv preprint arXiv:1508.00021*, 2015.

[21] J. J. V. Díaz, A. B. R. González, and M. R. Wilby, "Bluetooth traffic monitoring systems for travel time estimation on freeways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 123–132, 2016.

[22] T. Djukic, J. Barceló, M. Bullejos, L. Montero Mercadé, E. Cipriani, H. van Lint, and S. Hoogendoorn, "Advanced traffic data for dynamic od demand estimation: The state of the art and benchmark study," in *TRB 94th Annual Meeting Compendium of Papers*, 2015, pp. 1–16.

[23] P. Duan, G. Mao, W. Liang, and D. Zhang, "A unified spatio-temporal model for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[24] P. Duan, G. Mao, C. Zhang, and S. Wang, "Starima-based traffic prediction with time-varying lags," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on.* IEEE, 2016, pp. 1610–1615.

[25] N. G. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Network delay tomography from end-to-end unicast measurements," in *Thyrrhenian Internatinal Workshop on Digital Communications.* Springer, 2001, pp. 576–595.

[26] N. G. Duffield and F. L. Presti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Transactions on Networking (TON)*, vol. 12, no. 6, pp. 978–992, 2004.

[27] X. Feng, J. Guo, B. Qin, T. Liu, and Y. Liu, "Effective deep memory networks for distant supervised relation extraction." pp. 4002–4008, 2017.

[28] N. Furstenau, M. Schmidt, H. Horack, W. Goetze, and W. Schmidt, "Extrinsic fabry-perot interferometer vibration and acoustic sensor systems for airport ground traffic monitoring," *IEE Proceedings-Optoelectronics*, vol. 144, no. 3, pp. 134–144, 1997.

[29] N. J. Garber and L. A. Hoel, *Traffic and highway engineering.* Cengage Learning, 2014.

[30] D. Geiger, P. Wells, P. Bugas-Schramm, L. Love, S. McNeil, D. Merida, M. D. Meyer, R. Ritter, K. Steudle, D. Tuggle *et al.*, "Transportation asset management in australia, canada, england, and new zealand," Tech. Rep., 2005.

[31] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," 2019.

[32] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 714–727, 2010.

[33] Y. Guessous, M. Aron, N. Bhouri, and S. Cohen, "Estimating travel time distribution under different traffic conditions," *Transportation Research Procedia*, vol. 3, pp. 339–348, 2014.

[34] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, 2015.

[35] D. Hale, "An efficient method for computing local cross-correlations of multidimensional signals," in *CWP-544: Consortium Project on Seismic Inverse methods for Complex Structures*. Center for Wave Phenomena, Colorado, 2006, pp. 253–260.

[36] J. Han, J. W. Polak, J. Barria, and R. Krishnan, "On the estimation of space-mean-speed from inductive loop detector data," *Transportation planning and technology*, vol. 33, no. 1, pp. 91–104, 2010.

[37] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova, "Community detection in large-scale networks: a survey and empirical evaluation," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 6, pp. 426–439, 2014.

[38] E. J. Horvitz, J. Apacible, R. Sarin, and L. Liao, "Prediction, expectation,

and surprise: Methods, designs, and study of a deployed traffic forecasting service," *arXiv preprint arXiv:1207.1352*, 2012.

[39] J. Huang, X. Huangfu, H. Sun, H. Li, P. Zhao, H. Cheng, and Q. Song, "Backward path growth for efficient mobile sequential recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 46–60, 2015.

[40] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.

[41] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, 2017.

[42] F. Kelly, "The mathematics of traffic in networks," in *T. Gowers, editor, The Princeton companion to mathematics*, vol. 1, no. 1. Princeton University Press Princeton, 2008, pp. 862–870.

[43] J. P. Kennedy Jr, "Cellular based traffic sensor system," Nov. 7 1995, uS Patent 5,465,289.

[44] E. Kim, "Mrf model based real-time traffic flow prediction with support vector regression," *Electronics Letters*, vol. 53, no. 4, pp. 243–245, 2017.

[45] Y.-j. Kim, J.-s. Hong *et al.*, "Urban traffic flow prediction system using a multifactor pattern recognition model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2744–2755, 2015.

[46] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya, "Real-time measurement of link vehicle count and travel time in a road network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 814–825, 2010.

[47] Y. Lai, Z. Lv, K.-C. Li, and M. Liao, "Urban traffic coulomb's law: A new approach for taxi route recommendation," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[48] G. Leduc *et al.*, "Road traffic data: Collection methods and applications," *Working Papers on Energy, Transport and Climate Change*, vol. 1, no. 55, 2008.

[49] L. Li, X. Chen, Z. Li, and L. Zhang, "Freeway travel-time estimation based on temporal–spatial queueing model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1536–1541, 2013.

[50] R. Li, G. Rose, and M. Sarvi, "Using automatic vehicle identification data to gain insight into travel time variability and its causes," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1945, pp. 24–32, 2006.

[51] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[52] Z.-X. Li, X.-M. Yang, and Z. Li, "Application of cement-based piezoelectric sensors for monitoring traffic flows," *Journal of transportation engineering*, vol. 132, no. 7, pp. 565–573, 2006.

[53] F. Logi and S. G. Ritchie, "A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5-6, pp. 507–527, 2002.

[54] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 557–569, 2016.

[55] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[56] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[57] P. McGowen and M. Sanderson, "Accuracy of pneumatic road tube counters," vol. 1013, p. 2, 2011.

[58] X. Meng, "Scalable simple random sampling and stratified sampling," in *International Conference on Machine Learning*, 2013, pp. 531–539.

[59] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[60] N. Mitrovic, M. T. Asif, J. Dauwels, and P. Jaillet, "Low-dimensional models for compressed sensing and prediction of large-scale traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2949–2954, 2015.

[61] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.

[62] M. Mousa, E. Oudat, and C. Claudel, "A novel dual traffic/flash flood monitoring system using passive infrared/ultrasonic sensors," in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2015, pp. 388–397.

[63] E. K. Moylan and T. H. Rashidi, "Latent-segmentation, hazard-based models of travel time," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2174–2180, 2017.

[64] E. Namey, G. Guest, L. Thairu, and L. Johnson, "Data reduction techniques for large qualitative data sets," *Handbook for team-based qualitative research*, vol. 2, no. 1, pp. 137–161, 2008.

[65] R. P. D. Nath, H.-J. Lee, N. K. Chowdhury, and J.-W. Chang, "Modified k-means clustering for travel time prediction based on historical traffic data," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2010, pp. 511–521.

[66] H. Nguyen, W. Liu, and F. Chen, "Discovering congestion propagation patterns in spatio-temporal traffic data," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 169–180, 2017.

[67] W. H. Organization, *Global status report on road safety 2015*. World Health Organization, 2015.

[68] D. Pelleg, A. W. Moore *et al.*, "X-means: Extending k-means with efficient estimation of the number of clusters." in *Icml*, vol. 1, 2000, pp. 727–734.

[69] P. E. Pfeifer and S. J. Deutrch, "A three-stage iterative procedure for space-time modeling phillip," *Technometrics*, vol. 22, no. 1, pp. 35–47, 1980.

[70] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow

prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[71] A. Prokhorchuk, V. P. Payyada, J. Dauwels, and P. Jaillet, "Estimating travel time distributions using copula graphical lasso," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on.* IEEE, 2017, pp. 1–6.

[72] W. Pu, "Analytic relationships between travel time reliability measures," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2254, pp. 122–130, 2011.

[73] L. Qu, L. Li, Y. Zhang, and J. Hu, "Ppca-based missing data imputation for traffic flow volume: a systematical approach," *IEEE Transactions on intelligent transportation systems*, vol. 10, no. 3, pp. 512–522, 2009.

[74] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, "A cost-effective recommender system for taxi drivers," pp. 45–54, 2014.

[75] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[76] M. Rahmani, E. Jenelius, and H. Koutsopoulos, "Non-parametric estimation of route travel time distributions from low-frequency floating car data," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 343–362, 2015.

[77] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos, "Floating car and camera data fusion for non-parametric route travel time estimation," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on.* IEEE, 2014, pp. 1286–1291.

[78] M. Ramezani and N. Geroliminis, "On the estimation of arterial route travel

time distribution with markov chains," *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1576–1590, 2012.

[79] A. Salamanis, D. D. Kehagias, C. K. Filelis-Papadopoulos, D. Tzovaras, and G. A. Gravvanis, "Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1678–1687, 2016.

[80] I. Sanaullah, M. Quddus, and M. Enoch, "Developing travel time estimation methods using sparse gps data," *Journal of Intelligent Transportation Systems*, vol. 20, no. 6, pp. 532–544, 2016.

[81] D. W. Scott and G. R. Terrell, "Biased and unbiased cross-validation in density estimation," *Journal of the american Statistical association*, vol. 82, no. 400, pp. 1131–1146, 1987.

[82] S. C. Sen-Ching and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Visual Communications and Image Processing 2004*, vol. 5308. International Society for Optics and Photonics, 2004, pp. 881–892.

[83] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annual reviews in control*, vol. 43, pp. 128–151, 2017.

[84] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," *arXiv preprint arXiv:1902.09130*, 2019.

[85] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[86] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.

[87] S. L. Skszek, "State-of-the-art report on non-traditional traffic counting methods," Arizona. Dept. of Transportation, Tech. Rep., 2001.

[88] J. Sochor, A. Herout, and J. Havel, "Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3006–3015.

[89] L. Song, "Improved intelligent method for traffic flow prediction based on artificial neural networks and ant colony optimization." *Journal of Convergence Information Technology*, vol. 7, no. 8, 2012.

[90] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: video surveillance of highway traffic," in *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. ACM, 2004, pp. 131–135.

[91] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 2, pp. 121–135, 2003.

[92] A. Steed and R. Abou-Haidar, "Partitioning crowded virtual environments," pp. 7–14, 2003.

[93] N. B. Stoll, T. Glick, and M. A. Figliozzi, "Using high-resolution bus gps data to visualize and identify congestion hot spots in urban arterials," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2539, pp. 20–29, 2016.

[94] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[95] S. Susilawati, M. A. Taylor, and S. V. Somenahalli, "Distributions of travel time variability on urban roads," *Journal of Advanced Transportation*, vol. 47, no. 8, pp. 720–736, 2013.

[96] K. Tang, S. Chen, and Z. Liu, "Citywide spatial-temporal travel time estimation using big and sparse trajectories," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[97] Y. Tsang, M. Coates, and R. D. Nowak, "Network delay tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2125–2136, 2003.

[98] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1156–1166, 2012.

[99] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[100] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 672–683, 2009.

[101] A. Ventresque, Q. Bragard, E. S. Liu, D. Nowak, L. Murphy, G. Theodoropoulos, and Q. Liu, "Spartsim: a space partitioning guided by road network for distributed traffic simulations," pp. 202–209, 2012.

[102] T. Verma, P. Varakantham, S. Kraus, and H. C. Lau, "Augmenting decisions of taxi drivers through reinforcement learning for improving revenues," pp. 18–23, 2017.

[103] K. Wan, "Estimation of travel time distribution and travel time derivatives," Ph.D. dissertation, Princeton University, 2014.

[104] R. Wang, C.-Y. Chow, Y. Lyu, V. C. Lee, S. Kwong, Y. Li, and J. Zeng, "Taxirec: recommending road clusters to taxi drivers using ranking-based extreme learning machines," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 3, pp. 585–598, 2018.

[105] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 25–34.

[106] D. Weyns, T. Holvoet, and A. Helleboogh, "Anticipatory vehicle routing using delegate multi-agent systems," pp. 87–93, 2007.

[107] B. Williams, "Multivariate vehicular traffic flow prediction: evaluation of arimax modeling," *Journal of the Transportation Research Board*, no. 1776, pp. 194–200, 2001.

[108] B. Williams, P. Durvasula, and D. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1644, pp. 132–141, 1998.

[109] J. Wolff, T. Heuer, H. Gao, M. Weinmann, S. Voit, and U. Hartmann, "Parking monitor system based on magnetic field senso," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 1275–1279.

[110] D. Woodard, G. Nogin, P. Koch, D. Racz, M. Goldszmidt, and E. Horvitz, "Predicting travel time reliability using mobile phone gps data," *Transportation Research Part C: Emerging Technologies*, vol. 75, pp. 30–44, 2017.

[111] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on.* IEEE, 2008, pp. 5062–5068.

[112] Y.-J. Wu, F. Chen, C. Lu, B. Smith, and Y. Chen, "Traffic flow prediction for urban network using spatio-temporal random effects model," in *91st Annual Meeting of the Transportation Research Board (TRB)*, 2012.

[113] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial–temporal weighted model on mapreduce for short-term traffic flow forecasting," *Neurocomputing*, vol. 179, pp. 246–263, 2016.

[114] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

[115] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," pp. 25–36, 2012.

[116] Y. Xie, Y. Zhang, and Z. Ye, "Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 5, pp. 326–334, 2007.

[117] J. Xu, R. Rahmatizadeh, L. Boloni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2572–2581, 2018.

[118] Q. Yang, G. Wu, K. Boriboonsomsin, and M. Barth, "A novel arterial travel time distribution estimation model and its application to energy/emissions estimation," *Journal of Intelligent Transportation Systems*, pp. 1–13, 2017.

[119] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," 2018.

[120] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[121] J. Yeon, L. Elefteriadou, and S. Lawphongpanich, "Travel time estimation on a freeway using discrete time markov chains," *Transportation Research Part B: Methodological*, vol. 42, no. 4, pp. 325–338, 2008.

[122] T. Yi and B. M. Williams, "Dynamic traffic flow model for travel time estimation," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2526, pp. 70–78, 2015.

[123] U. Yildirim and Z. Çataltepe, "Short time traffic speed prediction using data from a number of different sensor locations," in *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on.* IEEE, 2008, pp. 1–6.

[124] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[125] X. Yu, S. Gao, X. Hu, and H. Park, "A markov decision process approach to vacant taxi routing with e-hailing," *Transportation Research Part B: Methodological*, vol. 121, pp. 114–134, 2019.

[126] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," pp. 109–118, 2011.

[127] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 10, pp. 2390–2403, 2013.

[128] G. Zhang, R. P. Avery, and Y. Wang, "Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras," *Transportation Research Record*, vol. 1993, no. 1, pp. 138–147, 2007.

[129] R. Zhang, S. Newman, M. Ortolani, and S. Silvestri, "A network tomography approach for traffic monitoring in smart cities," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[130] X. Zhang, Z. Zhao, Y. Zheng, and J. Li, "Prediction of taxi destinations using a novel data embedding method and ensemble learning," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[131] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 301–312.

[132] K. Zhao, D. Khryashchev, J. Freire, C. Silva, and H. Vo, "Predicting taxi demand at high spatial resolution: Approaching the limit of predictability," pp. 833–842, 2016.

[133] X. Zhou, H. Rong, C. Yang, Q. Zhang, A. V. Khezerlou, H. Zheng, M. Z. Shafiq, and A. X. Liu, "Optimizing taxi driver profit efficiency: A spatial network-based markov decision process approach," *IEEE Transactions on Big Data*, 2018.

[134] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.