A METHODOLOGY WITH DISTRIBUTED ALGORITHMS FOR LARGE-SCALE HUMAN MOBILITY PREDICTION

by

QiuLei Guo

B.S., South China University of Technology, China, 2010

M.S., South China University of Technology, China, 2013

Submitted to the Graduate Faculty of

the School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

This dissertation was presented

By

QiuLei Guo

It was defended on

Nov 03, 2017

and approved by

Hassan A. Karimi, Professor, School of Computing and Information, University of Pittsburgh

Balaji Palanisamy, Assistant Professor, School of Computing and Information, University of Pittsburgh

Paul Munro, Associate Professor, School of Computing and Information, University of Pittsburgh

ChaoWei Phil Yang, Professor, Department of Geography and GeoInformation Sciences, George Mason University

Zhen (Sean) Qian, Assistant Professor, Department of Civil and Environmental Engineering, Carnegie Mellon University

Thesis Director/Dissertation Advisor: Hassan A. Karimi, Professor, School of Computing and Information, University of Pittsburgh

A METHODOLOGY WITH DISTRIBUTED ALGORITHMS FOR LARGE-SCALE HUMAN MOBILITY PREDICTION

QiuLei Guo, PhD

University of Pittsburgh, 2017

In today's era of big data, huge amounts of spatial-temporal data related to human mobility, e.g., vehicle trajectories, are generated daily from all kinds of city-wide infrastructures. Understanding and accurately predicting such a large amount of spatial-temporal data could benefit many real-world applications, e.g., efficient transportation resource relocation. However, the mix of spatial and temporal patterns among these activities and the scale of the data (in a city level) pose great challenges for accurate predictions under real-time constraints.

To bridge the gap, this dissertation proposes a methodology for the prediction of large-scale human mobility, especially a city level's vehicle trajectory distribution across the road network. The thesis has several major components: (1) a novel model for the prediction of spatial-temporal activities such as people's outflow/inflow movements combining the latent and explicit features; (2) different models for the simulation of corresponding flow trajectory distributions in the road network, from which hot road segments and their formation can be predicted and identified in advance; (3) different

MapReduce-based distributed algorithms for the simulation and analysis of large-scale trajectory distributions under real-time constraints.

First, our proposed methodology quantifies the latent features of spatial and temporal factors through tensor factorization, given existing mobility datasets. We model the relationship between spatial-temporal activities and the latent and other explicit features as a Gaussian process, which can be viewed as a distribution over the possible functions to predict human mobility.

After the prediction of overall inflow/outflow, we further model these movements' trajectory distributions in the road network, from which the corresponding hot road segments and the possible causes, among other things, can be predicted in advance. For example, based on prediction, in the next half hour, a high percentage of vehicles that travel from region A/B toward region C/D might pass through the same road segment, which indicates a possible traffic jam/bottleneck there. This process is computationally intensive and requires efficient algorithms for real-time response because the scale of a city's road network and the possible number of trajectories that people might take during certain time periods could be very large. Efficient distributed algorithms are proposed and validated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

A large amount of spatial-temporal data related to human mobility accumulates daily from all kinds of city infrastructures, because of the rapid development and common use of location-sensing technologies, such as GPS and RFID sensors. Solving many real-world problems requires understanding and correctly predicting these spatial-temporal activities (for example, the outflow/inflow of people), as well as these movements' trajectory distributions in the road network. For example, by predicting the number of people who would leave or enter certain neighborhoods during the next half hour, taxi companies or Uber can optimally allocate their vehicles. Correspondingly, traffic agencies could further investigate and simulate these vehicle movements' corresponding trajectories in the road network and find the set of hot road segments with high centrality where lots of vehicles would pass by, from which future traffic congestions and their possible causes, among other things, can be predicted even before it happens. For example, based on the prediction, a high percentage of vehicles that travel from region A/B heading to region C/D might pass the same route in the next half hour, which would indicate a possible traffic jam or bottleneck there later—and as a result, we could send suggestions to some of those drivers to avoid this route if possible.

These problems pose many technical challenges. First, in order to predict spatial-temporal activities (for example, people's outflow/inflow in the urban environment), one natural approach is to identify both the spatial and temporal features of these activities and use these features to train a predictive model for future prediction. However, the mix of spatial and temporal patterns among human activities makes it difficult to identify and extract the spatial and temporal features, respectively, from existing mobility datasets. By assuming overall spatial and temporal closeness, many existing techniques use the information from adjacent spatial areas and recent time periods as the spatial and temporal features for prediction (Williams and Hoel 2003, Froehlich, Neumann et al. 2009, Kaltenbrunner, Meza et al. 2010, Chen, Hu et al. 2011, Nishi, Tsubouchi et al. 2014). However, there are a few problems with such methodologies. For example, there is no definition of how close two areas should be to one another in order to share a similar pattern, and also, close areas do not necessarily share a similar pattern. Existing works have similar problems with temporal characteristics. At the same time, it is difficult for these exiting methods to inherently take both spatial and temporal characteristics into consideration, given that spatial and temporal features have different scales and that there are unknown relationships between them and human mobility.

As for the second problem (the simulation of corresponding movements' trajectory distribution in the road network and the detection of hot road segments with high centrality), it poses many technical challenges in the areas of uncertainty and big data. First, we would need to accurately predict the flow of people across neighborhoods.

To infer their corresponding trajectory distributions in the road network, we would need to know how many people leave a place and their probable trajectories. However, considering that there are usually multiple routes from which people can choose from one place to another, it is hard to tell which route people might follow and/or the corresponding possibilities of them following each particular route. Besides this overall uncertainty, the scale of a city's road network and the number of trajectories that people usually take during certain time periods could be quite large. Take New York City as an example. There are 388,409 road intersections and 523,442 road segments (OpenStreetMap 2017). In 2001, people made approximate 209 million vehicles trips (a trip by a single privately operated vehicle) and traveled 3 billion vehicle miles (one vehicle mile of travel is the movement of one privately operated vehicle for one mile, regardless of the number of people in the vehicle) (Patricia S. Hu 2001). As for taxi cabs (one of the most important transportation modes in New York City), each day they carry over one million passengers and make, on average, 500,000 trips—adding up to 170 million trips during 2011 (Ferreira, Poco et al. 2013). These numbers indicate that the task of predicting a city level's trajectory distribution is computationally intensive and would require efficient algorithms for real-time responses.

To tackle these challenges, this dissertation proposes a comprehensive methodology for the prediction of large scale of human spatial-temporal mobility, especially a city level's trajectory distributions in the road network. An overview of our

methodology is given in Figure 1.1. Specifically, our methodology comprises several specific components.

First, we propose a novel methodology for prediction of spatial-temporal activities (such as human outflow/inflow and their corresponding destination/origin distribution) using the latent spatial and temporal features extracted through tensor factorization, given historical mobility datasets. One major motivation behind our methodology is that we suspect the patterns of many spatial-temporal activities, such as human mobility, are highly correlated to or dependent on the characteristics of spatial environments, temporal periods, and other factors. For example, residential neighborhoods and office districts have high volumes of outflow and inflow in the morning and in the evening, respectively. While this is an interesting observation analyzed qualitatively, it is not sufficient to allow for any prediction, such as the number of people who would be leaving/entering a residential neighborhood during certain time periods. With our proposed methodology, we can use this simple initial qualitative information to predict various spatial-temporal activities. In particular, we first identify and quantify the latent characteristics of different spatial environments and temporal factors through tensor factorization. Next, we propose to model the hidden relationship between spatial-temporal activity and extract latent features as a Gaussian process, which can be viewed as a distribution over the possible functions. One major advantage of this proposed methodology is that it inherently considers both spatial and temporal data characteristics. In particular, through mathematically modeling the characteristics of

different spatial areas, different time periods, and their relationship to mobility patterns as a Gaussian process, predictions can be made using the data from not only one specific spatial area or temporal time period of interest, but also from other areas and time periods with similar patterns.

After predicting the flow of people between neighborhoods, we further investigated and simulated those movements' corresponding trajectories in the road network, from which we could predict some important phenomenon, for example, finding a set of road segments that many vehicles would use and identify the causes or reasons for their heavy use, such as the origins or destinations of the majority of the traffic in those road segments. Given that there are usually multiple routes that people can choose to go from one place to another, there is a challenge of uncertainty. Some previous works (Matthias and Zuefle 2008, Ren, Ercsey-Ravasz et al. 2014, Deri and Moura 2015) assumed people always choose the shortest paths. However, this might not be the case since people seldom strictly follow the shortest paths in their daily driving. To bridge the gap, we propose several models of vehicles' trajectory distributions in the road network, such as one based on the multivariate kernel density estimation. We provided a case study of Beijing's taxi data and compared our proposed models with traditional models, such as the shortest path. Experimental results demonstrate the advantage of our proposed model.

It is worth pointing out that the problems discussed above are very computationally intensive when considering the scale of a city's road network and the

numerous trajectories that people might take during a certain time period. With the advent of emerging cloud technologies, a natural and cost-effective approach to manage such large-scale data is to store them in a cloud environment and process them using modern distributed computing paradigms, such as MapReduce (Dean and Ghemawat 2008). In this work, different MapReduce-based distributed algorithms are proposed for (1) simulating vehicle trajectory distributions in the road network, based on the predicted outflow/inflow movements between neighborhoods from the previous step; and (2) analyzing the synthetic large-scale trajectory distributions in order to find interesting phenomena, such as the road segments that many vehicles might use, as well as the causes of these phenomena, like the origin and destinations of the majority of the traffic.

It should be pointed out that a trajectory is a unique way to represent people's spatial-temporal activity. It can be viewed as a sequence of time-ordered location records, such as a series of GPS points with latitude and longitude, or as a sequence of connected road segments in the road network. There are many techniques developed to predict a single vehicle's future trajectory, based on its initial partial trajectory (Liu and Karimi 2006, Froehlich and Krumm 2008, Chen, Lv et al. 2010, Jeung, Yiu et al. 2010). One major difference between these existing works and the proposed work in this thesis is that we focus more on people's/vehicle's movements at a city level and the corresponding trajectory distributions, instead of on a single vehicle's personal routing preference in the road network, based on its partial initial trajectory and history patterns. For several reasons, these personal predictions cannot be aggregated to achieve a city-level prediction.

First, the mobility problem addressed in this paper is quite different from those that have been addressed in previous works. In particular, most existing works seek to answer the question: Given a partial initial trajectory of a vehicle already in the road network, what is its most likely future trajectory in the road network? However, our methodology tries to answer the questions: How many people are heading from one specific neighborhood to another in the near future, say in the next hour?; What are the probable trajectories of these movements?; Which road segments would have a high degree of centrality (a lot of vehicles would pass by) and result in traffic jams?; and What are the origins and destinations of the traffic that passes through those hot road segments? Besides, due to privacy and technical issues, it is difficult to collect and store everyone's trajectory at the necessary level of detail (such as every two minutes) at the city level. On the other hand, some mobility datasets with less detail (namely, those with only origin and destination information for each trip) are more widely available, such as the census data/travel survey (Jiang, Ferreira Jr et al. 2012), mobile phone records (Gao, Liu et al. 2013), check-ins from location-based social networks such as Foursquare (Wei, Zheng et al. 2012), and others. Our proposed methodology is flexible and can properly handle both cases. Finally, the scale of the problem (a city-level trajectory distribution computation) is computationally intensive and requires efficient distributed algorithms to achieve suitable performance.

There are also some other related works, such as those that include the discovery of popular trajectories or hot routes from historical datasets (Li, Han et al. 2007, Chen,

Shen et al. 2011, Wei, Zheng et al. 2012, Han, Liu et al. 2015) and an estimation of the current traffic situation from Twitter (Sayyadi, Hurst et al. 2009, Castro, Zhang et al. 2012, Chen, Chen et al. 2014, Liu, Fu et al. 2014, Wang, Li et al. 2016). While these proposed techniques can find some interesting phenomena, such as popular routes and traffic jams that have previously happened or that are happening at the moment, they provide little assistance to future predictions. For example, there could be a local event in a neighborhood today with several road segments blocked by the police, which would cause some of the nearby roads to be congested with a higher traffic volume than usual—or maybe not, depending on people's mobility at that time and the nearby road network topology. Mining historical hot routes cannot predict these abnormal situations. On the other hand, with the proposed methodology in this work, we can predict people's flow volume across neighborhoods at a city level, simulate their corresponding trajectories in the road network by blocking corresponding road segments, and check to see if any nearby road segments would become crowded or remain clear.

The proposed methodology in this paper could also shed light on a future Intelligent Transportation System prototype that would help alleviate traffic congestion problems in metropolitan cities. Specifically, as self-driving vehicles become feasible and even prevalent in the future, our methodology could be used in a public cloud environment, where self-driving vehicles on the road network would act as the clients and send their movement information to the cloud in advance, including both their origins and destinations. The cloud would aggregate this information, estimate the trajectory

distribution in the road network based on the routing strategies of self-driving vehicles, and detect the corresponding levels of traffic. If a congestion is predicted (too many vehicles would try to use the same route in the near future), the cloud would send this information to affected self-driving vehicles so that they could update their routes (choose less crowded routes).

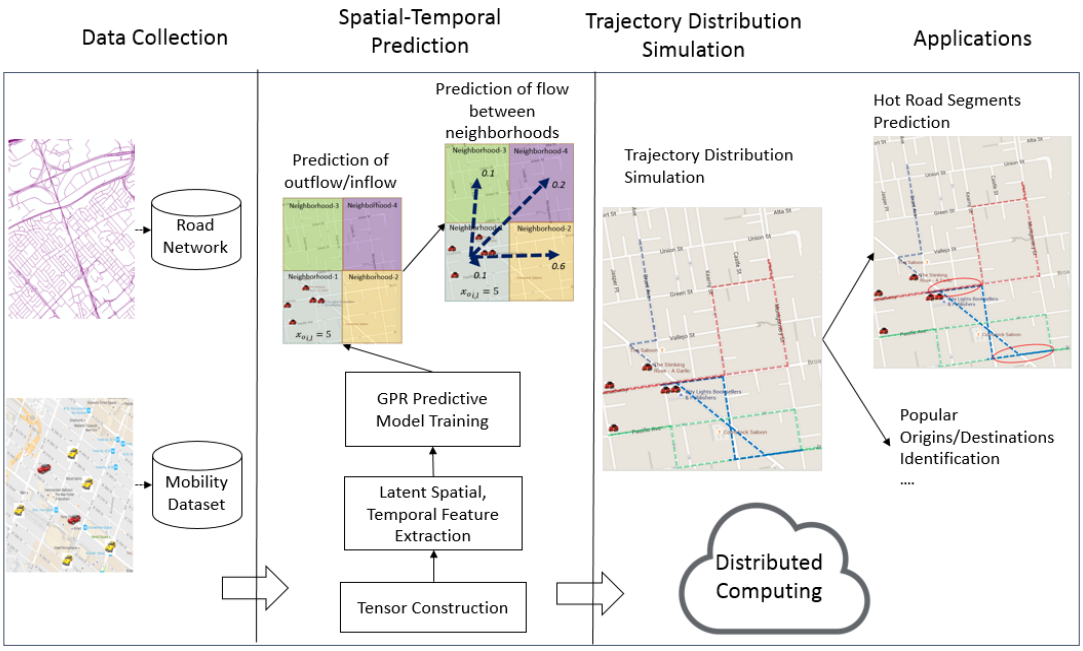


Figure 1.1 an overview of the proposed methodology

1.1 Research Problems

This thesis tackles the challenges of the prediction of human mobility on a large scale. In particular, we focus on people's spatial-temporal mobility of outflow/inflow, and their trajectory distributions in the road network, from which we could optimally reallocate transportation resources, such as taxis or Uber vehicles, and estimate future traffic situations, such as congestion and its possible causes, among others. In particular, this research addresses the following questions:

1. How can we quantify the features of the spatial and temporal factors, based on the existing mobility dataset?
2. How can we mathematically model the relationship between the extracted spatial-temporal features and people's mobility, such as outflow/inflow in an urban environment, for future predictions?
3. How can we accurately model people's trajectory distributions in the road network based on the previous predicted flows?
4. How can we efficiently simulate the huge amount of movement trajectory distributions in a city level's road network?
5. How can we efficiently process the large scale of trajectory distributions generated from previous steps for some useful information, such as predicting the

set of hot road segments and identifying where the majority of traffic in those road segments are coming from or going to?

1.2 Contributions

The research in this thesis has six major contributions:

(1) A comprehensive methodology for the prediction of people's mobility at a large scale.

(2) A novel model to predict spatial-temporal activity using latent spatial-temporal features extracted from existing mobility data.

(3) Different models for the estimation of vehicle trajectory distributions in a road network.

(4) A distributed algorithm for the real-time simulation of large-scale trajectory distributions in a road network.

(5) Different distributed algorithms for the processing and analysis of large-scale trajectory distribution, such as the prediction of hot road segments that are based on such analyses.

(6) Case studies based on real-world data collected from New York City and Beijing's taxi trip data sets.

1.3 Chapters Overview

The rest of the proposal is organized as follows. Section 2 reviews background information and related work. Section 3 presents the proposed novel methodology for the prediction of human spatial-temporal mobility, using latent features. Section 4 presents the models of trajectory distributions in the road network. Section 5 provides different MapReduce-based distributed algorithms, including the simulation of the corresponding trajectory distributions in the road network and the analysis of the simulated trajectory distributions, such as the prediction of hot road segments. Section 6 conducts case studies with data sets of taxi trips taken in both New York City and Beijing, and systematically evaluates our proposed methodology. Section 7 provides the conclusions of this thesis and future research direction.

2.0 BACKGROUND AND RELATED WORK

Issues of human mobility have attracted lots of attention for a long time from researchers in a wide variety of fields, such as urban planning, sociology, computer science, and geology, among others. This chapter reviews how existing work analyzes and predicts human spatial-temporal activities from different perspectives, their limitations, and the difference between them and the proposed work in this thesis.

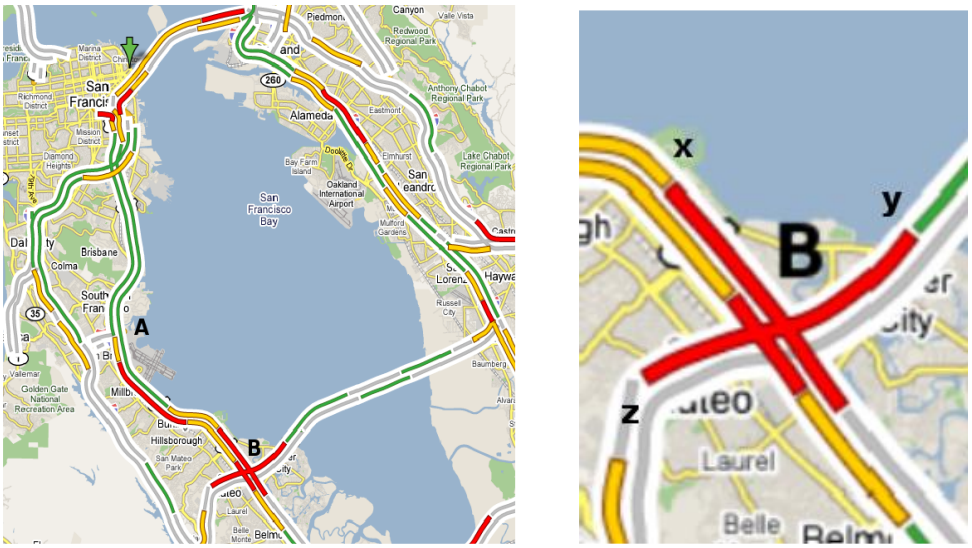
2.1 Traffic Prediction

Traditionally, researchers have used static models, such as the gravity model (Wilson 1967), to estimate the amount of interactions between two geographic areas, such as two cities. With the invention of some infrastructure sensors, such as a traffic loop that can count the number of vehicles passing a road segment, these models have been widely deployed in cities' road networks. Many models have been developed to predict the traffic situation from these data. Davis and Nihan (Davis and Nihan 1991) suggested a nonparametric k-nearest neighborhood approach to predict short-term traffic volume. The general idea is to use the recent traffic volume from a to-be predicted freeway and its

adjacent freeways as the input vector, to find the top-k closest vectors in history, and compute the average value. Clark (Clark 2003) proposed a similar k-NN approach, but with more input variables and different outputs; besides the traffic volume, this model also collects and predict the speed, flow, occupancy, and other factors, as well as explores the accuracy between different univariate or multivariate models. Williams and Hoel (Williams and Hoel 2003) presented the theoretical basis for modeling univariate traffic condition data streams as seasonal autoregressive integrated moving average processes. Shekhar and Williams (Shekhar and Williams 2008) presented an adaptive parameter estimation methodology for univariate traffic condition forecasting through the use of three well-known filtering techniques: the Kalman filter, recursive least squares, and least mean squares.

One limitation of these works is that they can only predict the traffic volume of a single road segment in isolation, and cannot provide any other information, such as the causes of possible traffic jams or the patterns of people's mobility at a higher level, leaving the question open as to where the traffic in those road segments is coming from or where it is going. This information would help traffic agencies optimize the traffic resource more efficiently. Figure 2.1 (Li, Han et al. 2007) gives a good example of this issue. It shows traffic data in the San Francisco Bay Area on a weekday at approximately 7:30 am local time. Different colors show different levels of congestion (for example, dark red shows heavy congestion). We can see that there are some congestions in the road network, but we do not know why this congestion is occurring. If we can predict that

traffic jams are formed because many people are driving from location Y to location X, the traffic agencies could increase the frequency of corresponding public buses traveling from Y to X during those time periods to reduce the volume of private traffic.



(a) The Bay Area (b) A closer look at the congested area

Figure 2.1 Snapshots of San Francisco traffic

Besides these limitations, the high cost of deploying and maintaining the infrastructure of traffic loops also limits their coverage. Motivated by the popularity of location-based applications and social networks such as Twitter, many recent studies have been conducted to explore these social media data for its use in estimating traffic situations. The core idea of this field is to detect traffic-related tweets and use them to

estimate the current traffic situation. Sayyadi et al. (Sayyadi, Hurst et al. 2009) proposed and developed an event-detection algorithm which creates a keyword graph and uses community detection methods analogous to those used for social network analysis to discover and describe events. Liu et al. (Liu, Fu et al. 2014) presented an application for traffic event detection and summaries, based on mining representative terms from the tweets posted when anomalies occur. Chen et al. (Chen, Chen et al. 2014) presented a unified statistical framework that combines two models based on hinge-loss Markov random fields (HLMRFs) to monitor traffic congestion through feeds from tweet streams.

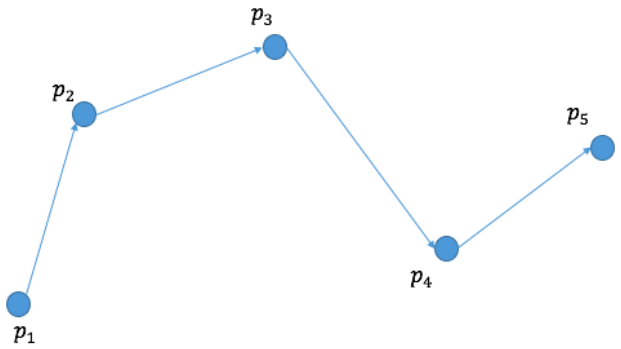
Although using crowd-sourced data from social networks have some advantages in some cases, these existing methodologies also have limitations such as failing to detect many ongoing traffic events, due to the sparsity of traffic-related information on social networks (since few people are likely to tweet about the traffic situation while driving) and they also gain little insight of people's travelling patterns. In addition to these limitations, the proposed technique in this thesis and the works above also have different foci. Those works previously cited focus more on the estimation of the current traffic situation through extracting the traffic-related information from the tweets that people posted about their current traffic situations. However, our proposed methodology focuses more on the prediction of future movements; people's outflow/inflow across neighborhoods, their corresponding possible trajectory distribution in the road network, and the set of hot road segments where lots of vehicles might pass by in the near future.

There are also some other related works such as the abnormal spatial events detection, e.g., people's gathering events. (Neill 2009) proposed a two-step approach based on the expectation-based scan statistic for the detection of emerging spatial patterns through monitoring a large number of spatially localized time series. (Hong, Zheng et al. 2015) modeled human mobility as Spatio-Temporal Graph (STG) for the detection of phenomena, entitled black holes and volcanos. Specifically, a black hole is a subgraph (of STG) that has the overall inflow greater than the outflow by a threshold while volcanos is the other way around. (Zhou, Khezerlou et al. 2016) proposed a model of Gathering directed acyclic Graph (G-Graph) for the early detection of gathering events. To improve the computation efficiency, they also designed an algorithm called SmartEdge.

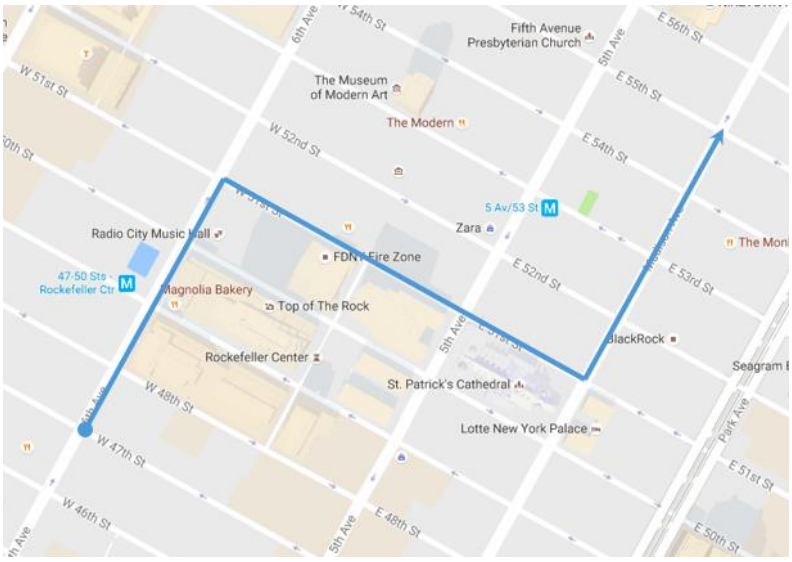
Apart from vehicles' traffic in the road network, there are also some studies on other modes of transportation or urban activity such as pedestrians, shared bicycle system, etc. Nishi et al. (Nishi, Tsubouchi et al. 2014) described a statistic-based method to estimate trends in the pedestrian population using location data collected from Yahoo! Japan app users. Froehlich et al. (Froehlich, Neumann et al. 2009) provided a spatial-temporal analysis of bicycle station usage in Barcelona and compared experimental results from four simple predictive models. Kaltenbrunner et al. (Kaltenbrunner, Meza et al. 2010) also provided spatial-temporal analysis for bicycle usage in Barcelona and adopted an autoregressive-moving-average (ARMA) model to predict the number of bikes and docks available at each bike station.

2.2 Trajectory Mining

The pervasive use of location-sensing technology such as GPS receivers and WiFi embedded in mobile devices has led to the accumulation of huge amounts of trajectory data. Generally, a trajectory can be viewed as a sequence of data points with location information (Figure 2.2a) or as road segments (Figure 2.2b).



(a) Trajectory of data points



(b) Trajectory of road segments

Figure 2.2. Illustrations of trajectory data

2.2.1 Individual Trajectory Predictions

Among the various topics in the field of trajectory mining, predicting the future trajectory of a person or vehicle is of great interest. Liu and Karimi (Liu and Karimi 2006) presented two models for trajectory prediction: a probability-based model and a learning-based model. Froehlich and Krumm (Froehlich and Krumm 2008) developed the algorithms for predicting the end-to-end route of a vehicle, mainly based on GPS observations of the vehicle's past trips. Jeung et al. (Jeung, Yiu et al. 2010) presented a maximum likelihood and a greedy algorithm for predicting the travel path of an object, based on a developed mobility model that offers a concise representation of mobility

statistics extracted from massive collections of historical object trajectories. Scellato et al. (Scellato, Musolesi et al. 2011) created a spatial-temporal location prediction model for a single user, based on his/her own historical trajectories. Zhang et al. (Zhang, Lin et al. 2016) introduced EigenTransitions, a spectrum-based, generic framework for analyzing mobility datasets and predicting an individual user's mobility, such as the next area they are likely to visit. As discussed above, the major application of these studies was to predict a single vehicle's personal routing preference in the road network, based on its partial initial trajectory and history patterns. On the other hand, the proposed work in this thesis focuses on people's movements at a city level and their corresponding trajectory distributions, which is computationally intensive. As a result, an efficiently distributed solution is needed. Furthermore, due to privacy and technical issues, it is difficult to frequently collect a series of GPS points from many individual users to gain an overview of a city level's mobility and the corresponding traffic situation in the near future, as with the input data required by these studies; in contrast, our methodology can handle some less detailed datasets, such as a huge number of anonymous trips with only origins, destinations, and their corresponding timestamps.

2.2.2 Popular Trajectory Mining

Mining popular routes from existing trajectory datasets is another topic that is close to our proposed methodology. Li et al. (Li, Han et al. 2007) proposed a density-based algorithm named FlowScan to cluster road segments based on the density of common

traffic they share. Zhu et al. (Zhu, Luo et al. 2010) proposed a novel three-phase approach to discover a tropical cyclone's trajectory corridors, based on clustering methods. Chen et al. (Chen, Shen et al. 2011) investigated the most popular route (MPR) between two locations by observing the traveling behaviors of many previous users. They developed an algorithm to retrieve a transfer network from raw trajectories that would indicate all the possible movements between locations. After that, the absorbing Markov chain model is applied to derive a reasonable transfer probability for each transfer node in the network. Comito et al. (Comito, Falcone et al. 2015) defined and implemented a novel methodology to mine popular travel routes from geo-tagged posts. Han et al. (Han, Liu et al. 2015) designed a road-network aware approach, named NEAT, for the fast and effective clustering of trajectories of mobile objects travelling in road networks. More specifically, NEAT can discover spatial clusters as groups of sub-trajectories that describe both dense and highly continuous flows of mobile objects.

Compared with our proposed methodology in this thesis, these existing techniques focus on mining phenomena such as popular routes or historical traffic jams, but cannot provide much information for future situations, especially when some of conditions change. For example, there might be a parade in a neighborhood this afternoon that would cause several road segments to be blocked by the police, which could lead to a drastic change in trajectory patterns. In order to estimate the overall impact of such an event, the city agencies can use our proposed methodology to predict people's movements and simulate the corresponding trajectory distributions by blocking those

road segments, so they could check if any of nearby road segments would become too crowded.

2.2.3 Other Trajectory Mining

Other studies have also been conducted to mine trajectory datasets to reveal different interesting urban activities. Guo et al. (Guo, Liu et al. 2010) developed a graph-based approach that converts trajectory data to a graph-based representation and treats it as a complex network, to which they further apply a spatially constrained graph partitioning method to discover natural regions defined by trajectories. Liu et al. (Liu, Liu et al. 2010) presented a novel, non-density-based approach called mobility-based clustering to identify hot spots of moving vehicles in an urban area. The key idea is to use the sample objects' instant mobility (taxi trajectory data) as the "sensors" to perceive the vehicle density in nearby areas. Liu et al. (Liu, Zhu et al. 2012) proposed a novel algorithm for recognizing urban roads with coarse-grained GPS traces from probe vehicles moving in urban areas. Zhang et al. (Zhang, Wilkie et al. 2013) proposed a step toward real-time sensing of refueling behavior and citywide fuel consumption using the reported trajectories from a fleet of GPS-equipped taxicabs. Wang et al. (Wang, Zheng et al. 2014) presented a citywide and real-time model for estimating the travel time of any path in real time in a city, based on the GPS trajectories of vehicles received in current time slots and over a period of history, as well as information from map data sources.

2.3 Urban Community and Event Analysis

In addition to the trajectory dataset, exploring and discovering hidden interesting phenomena based on other spatial-temporal datasets, such as location-based social networks, has also attracted much attention. Spatial community discovery/analysis is one of the hottest research topics, among others. Cranshaw et al. (Cranshaw, Schwartz et al. 2012) introduced a clustering model and research methodology for studying the structure and composition of a city on a large scale, based on the social media information that its residents generate. Noulas et al. (Noulas, Scellato et al. 2011) also proposed an approach to cluster geographic areas with similar categories. This study also clustered the users according to the types of places they check in and the frequency of check-ins. Yuan et al. (Yuan, Zheng et al. 2012) proposed a framework (titled DRoF) that discovers regions of different functions in a city, using both human mobility among regions and points of interests (POIs) located in a region.

Many other interesting phenomena have been explored besides the spatial community. Comito et al. (Comito, Falcone et al. 2015) proposed a methodology to infer interesting locations and frequent travel sequences among these locations in a given geo-spatial region from geo-tagged tweets. Kamath et al. (Kamath, Caverlee et al. 2012) explored how the factors of spatial influence and interest affinity affect the global spread of social media. Noulas and Mascolo (Noulas and Mascolo 2013) inferred the functions

of each neighborhood in the city by using Foursquare POIs and cellular data. Finally, Quercia et al. (Quercia, Aiello et al. 2015) explored the possibilities of using social media data from Flickr and Foursquare to automatically identify safe and walkable streets.

Other datasets, such as phone usage, census-based data, and public transportation records, among others, have also attracted much attention, in addition to location-based social networks. Lathia et al. (Lathia, Quercia et al. 2012) explored the correlation between London's urban flow of public transport and the well-being of London's census areas (measured by census-based indices), from which some phenomena are found, such as a segregation effect. Lam and Bouillet (Lam and Bouillet 2014) proposed an efficient real-time algorithm to cluster the events generated by the sensors available from traffic light control systems, which are composed of an induction loop which is triggered whenever a metallic object is detected, such as a car. Zheng et al. (Zheng, Liu et al. 2014) inferred the fine-grained noise situation at different times of day for each region of NYC by modeling the noise situation of NYC with a three-dimensional tensor and supplementing the missing entries of the tensor through a context-aware tensor decomposition approach. Finally, Liu et al. (Liu, Wang et al. 2012) derived urban land-use information by classifying the study area into six types of "source-sink" areas through taxi data on pick-ups and drop-offs in Shanghai.

2.4 Distributed Computing

Since the scale of many spatial-temporal datasets nowadays could be as large as tens of hundreds of gigabytes (or even larger), creating a real-time query and prediction method to use this large amount of data poses great challenges for a single commodity computer. As cloud computing has emerged as a cost-effective and promising solution for both computing- and data-intensive problems, a natural approach to manage such large-scale data is to store and process these datasets in a cloud service using modern distributed computing paradigms such as MapReduce.

2.4.1 MapReduce

MapReduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks (Dean and Ghemawat 2008). Hadoop is a popular open source implementation of the MapReduce framework. Hadoop is composed of two major parts: the storage model (the Hadoop distributed file system , or HDFS), and the compute model (MapReduce). Figure 2.3 shows an execution overview of the MapReduce model.

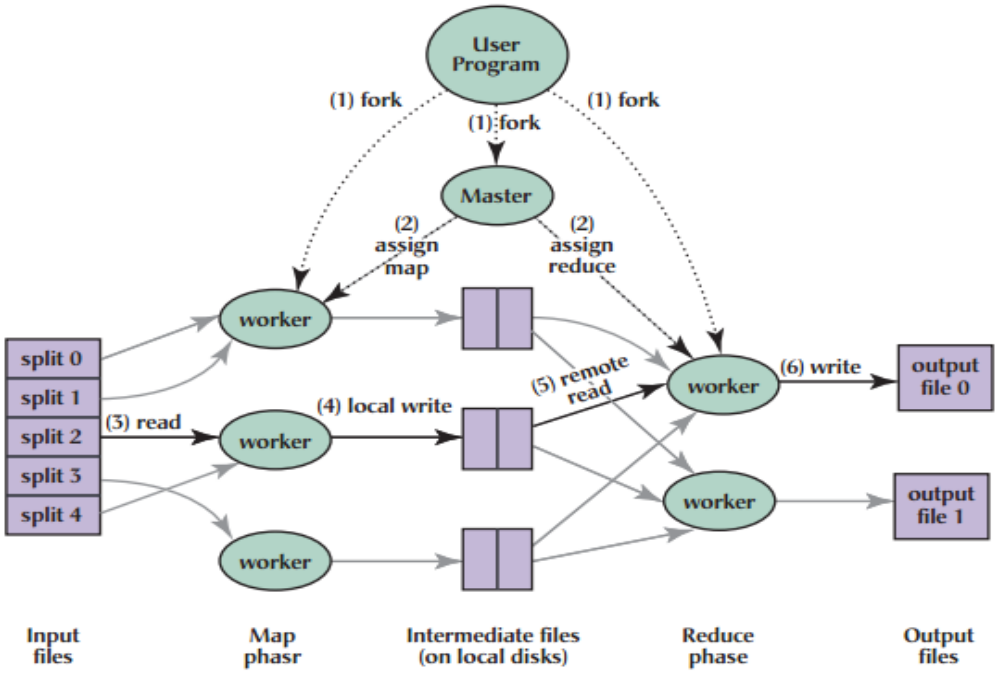


Figure 2.3 Execution overview of MapReduce model (Dean and Ghemawat 2008)

A key feature of the MapReduce framework is that it can distribute a large job into several independent maps, and reduce tasks over several nodes of a large data center and process them in parallel. At the same time, MapReduce can effectively leverage data locality and processing on or near the storage nodes, and results in faster execution of the jobs. The framework consists of one master node and a set of worker nodes. In the map phase, the master node schedules and distributes the individual map tasks to the worker nodes. A map task executed in a worker node processes the smaller chunk of the file stored in HDFS and passes the intermediate results to the appropriate reduce tasks that are being executed in a set of worker nodes. The reduce

tasks collect the intermediate results from the map tasks and combine/reduce them to form the final output. Since each map operation is independent of the others, all map tasks can be performed in parallel. The same process occurs with reducers, as each reducer works on a mutually exclusive set of intermediate results produced by mappers.

2.4.2 Spatial Data Processing in Hadoop

Since MapReduce/Hadoop has become the defacto standard for distributed computation on a massive scale, some recent works have developed several MapReduce-based algorithms for spatial problems. Puri et al. (Puri, Agarwal et al. 2013) proposed and implemented a MapReduce algorithm for distributed polygon overlay computation in Hadoop. Ji et al. (Ji, Dong et al. 2012) presented a MapReduce-based approach that constructs an inverted grid index and processes kNN query over large spatial data sets. Akdogan et al. (Akdogan, Demiryurek et al. 2010) designed a unique spatial index and Voronoi diagram for given points in 2D space, which enables the efficient processing of a wide range of geospatial queries, such as RNN, MaxRNN and kNN with the MapReduce programming model. (Guo, Palanisamy et al. 2014) developed a MapReduce-based parallel polygon retrieval algorithm which aims to minimize the IO and CPU loads of the map and reduce tasks during spatial data processing. Hadoop-GIS (Wang, Lee et al. 2011) and Spatial-Hadoop (Eldawy, Li et al. 2013, Eldawy and Mokbel 2013) are two scalable, high-performance spatial data processing systems for running large-scale spatial queries

in Hadoop. These systems provide support for some fundamental spatial queries, such as the minimal bounding box query.

However, these studies only support some static spatial queries. They do not support spatial-temporal trajectory predictions, simulations, and the corresponding discovery of hot road segments that are addressed in this thesis. As a result, we propose to devise specific optimization techniques for an efficient implementation of the parallel trajectory prediction and simulation functions in MapReduce.

3.0 NOVEL SPATIAL-TEMPORAL PREDICTION USING LATENT FEATURES

In this section, the spatial-temporal prediction methodology that uses the latent features will be presented in detail. First, we describe how to model people's spatial-temporal fluxes as a tensor and extract the latent spatial-temporal features through factorization. Then, we present how to mathematically model the relationship between those extracted latent features and human mobility using a Gaussian process regression for future prediction.

3.1 Tensor Model of the Spatial-Temporal Activities

A tensor is a multidimensional array. Decompositions of a higher-order tensor can be used to extract and explain the properties among the tensor, which have wide applications in computer vision, numerical analysis, data mining, neuroscience, graph analysis, and elsewhere (Kolda and Bader 2009). In this thesis, we propose to model human fluxes between different neighborhoods with a 3-dimensional tensor $\mathcal{H} \in \mathcal{R}^{N \times N \times L}$, as shown in Figure 3.2. The first dimension of the tensor $\mathcal{H}$ denotes N origin neighborhoods, the

second dimension denotes N destination neighborhoods, and the third dimension denotes L time slots, respectively. Each entry of the tensor $\mathcal{H}(i, j, l)$ stores the average number of trips starting from neighborhood i to neighborhood j during time period l.

With this tensor model, we extract the latent spatial features of each origin neighborhood, destination neighborhood, and the latent temporal feature of each time slot through a Tucker decomposition. The Tucker decomposition can be thought of as the form of higher-order Principal Component Analysis (PCA). It decomposes a tensor into a core tensor multiplied by a matrix along each dimension (Kolda and Bader 2009). In our case, we decompose the tensor $\mathcal{H}$ into three matrices ${\mathcal{S}_o}^{N\times P}$, ${\mathcal{S}_d}^{N\times Q}$, ${\mathcal{T}}^{L\times R}$, and a core tensor $G^{P\times Q\times R}$, respectively, as shown in Figure 4.3. Mathematically, this relationship can be expressed as in Equation 3.1:

$$\mathcal{H} \approx G \times_1 \mathcal{S}_o \times_2 \mathcal{S}_d \times_3 \mathcal{T} = \sum_p \sum_q \sum_r g_{pqr}\, \mathcal{S}_{o_{:,p}} \circ \mathcal{S}_{d_{:,q}} \circ \mathcal{T}_{:,r} \quad (3.1)$$

Each element $\mathcal{H}$ is:

$$h_{ijl} \approx \sum_p \sum_q \sum_r g_{pqr} \mathcal{S}_{o_{i,p}} \mathcal{S}_{d_{j,q}} \mathcal{T}_{l,r} \quad (3.2)$$

Here, the symbol "$\circ$" stands for the vector outer product, which means that each element of the tensor is the product of the corresponding vector elements. $\mathcal{S}_{o_{:,p}}$ indicates the p^{th} column of matrix $\mathcal{S}_o$ and $\mathcal{S}_{o_{i,p}}$ is the i^{th} element in the p^{th} column. $\mathcal{S}_o$, $\mathcal{S}_d$ and $\mathcal{T}$ are the factor matrices and can be viewed as the principal component of the tensor's three corresponding dimensions. In other words, the row i of matrix $\mathcal{S}_o$, $\mathcal{S}_{o_{i,:}}$, is the feature vector that indicates the characteristics of origin neighborhood i. Similarly, the row j of

matrix $\mathcal{S}_d$, $\mathcal{S}_{d_{j,:}}$, is the feature vector that indicates the characteristics of destination neighborhood j. $\mathcal{T}_{l,:,}$ is the feature vector that indicates the characteristics of the corresponding time slot l. Each entry of the core tensor G indicates the level of interaction among different components of $\mathcal{S}_o$, $\mathcal{S}_d$, and $\mathcal{T}$, respectively.

This decomposition problem can be turned into an optimization problem:

$$\min \quad ||\mathcal{H} - G \times_1 \mathcal{S}_o \times_2 \mathcal{S}_d \times_3 \mathcal{T}||^2 \quad (3.3)$$

$$\text{subject to } G \in \mathcal{R}^{P\times Q\times R},$$

$$\mathcal{S}_o \in \mathcal{R}^{N\times P},$$

$$\mathcal{S}_d \in \mathcal{R}^{N\times Q},$$

$$\mathcal{T} \in \mathcal{R}^{L\times R}$$

To solve this optimization problem, (De Lathauwer, De Moor et al. 2000) designed a higher-order orthogonal iteration algorithm. In our case, the algorithm is shown in Figure 3.1:

procedure HOOI($\mathcal{H}, P, Q, R$)
  $\mathcal{S}_o \leftarrow P$ leading left singular vectors of $\mathcal{H}_{(1)}$
    ▷ $\mathcal{H}_{(1)}$ is the matricization of the tensor $\mathcal{H}$ along the dimension-1.
  $\mathcal{S}_d \leftarrow Q$ leading left singular vectors of $\mathcal{H}_{(2)}$
  $\mathcal{T} \leftarrow R$ leading left singular vectors of $\mathcal{H}_{(3)}$
  $G \leftarrow \mathcal{H} \times_1 \mathcal{S}_o^T \times_2 \mathcal{S}_d^T \times_3 \mathcal{T}^T$
  repeat
    $\mathcal{Y} \leftarrow \mathcal{H} \times_2 \mathcal{S}_d^T \times_3 \mathcal{T}^T$
    $\mathcal{S}_o \leftarrow P$ leading left singular vectors of $\mathcal{Y}_{(1)}$
    $\mathcal{Y} \leftarrow \mathcal{H} \times_1 \mathcal{S}_o^T \times_3 \mathcal{T}^T$
    $\mathcal{S}_d \leftarrow Q$ leading left singular vectors of $\mathcal{Y}_{(2)}$
    $\mathcal{Y} \leftarrow \mathcal{H} \times_1 \mathcal{S}_o^T \times_2 \mathcal{S}_d^T$
    $\mathcal{T} \leftarrow R$ leading left singular vectors of $\mathcal{Y}_{(3)}$
  until all matrices become stable or maximum iterations exhausted
  $G \leftarrow \mathcal{H} \times_1 \mathcal{S}_o^T \times_2 \mathcal{S}_d^T \times_3 \mathcal{T}^T$
  return

Figure 3.1. Higher-order orthogonal iteration algorithm

The motivation behind using the tensor factorization is that we think the existence of some latent features and interactions among them usually determine the patterns of many spatial-temporal activities such as how people in one neighborhood (origin) move to another neighborhood (destination) during certain time periods. For example, two residential neighborhoods would both have a high volume of outflow (to an office district) in the morning. Similarly, two nightlife districts would both attract a high volume of inflow in the evening. This is a simple qualitative analysis that is difficult to extend to general cases, since most regions are not monofunctional and people's flow is usually a mix of a variety of life patterns. However, by discovering the latent features and the interactions among them, we can mathematically model people's movements with respect to a certain neighborhood during certain time periods for future prediction. This is

somewhat similar to the recommendation system like the one Netflix uses, where a multidimensional tensor represents how different users rate different movies under various contexts, such as different times. For example, two users might give a high rating to a certain movie if they both liked the actors/actresses in the movie, or if the movie was a romantic movie, which was preferred by both users in the previous couple of weeks. Hence, if we can discover these latent features, we should be able to predict a rating with respect to a certain user and a certain item under specific contexts. Similarly, given the extracted latent features of origin neighborhoods (like users), destination neighborhoods (like movies), the specific time period, and some other features, we could predict people's flow.

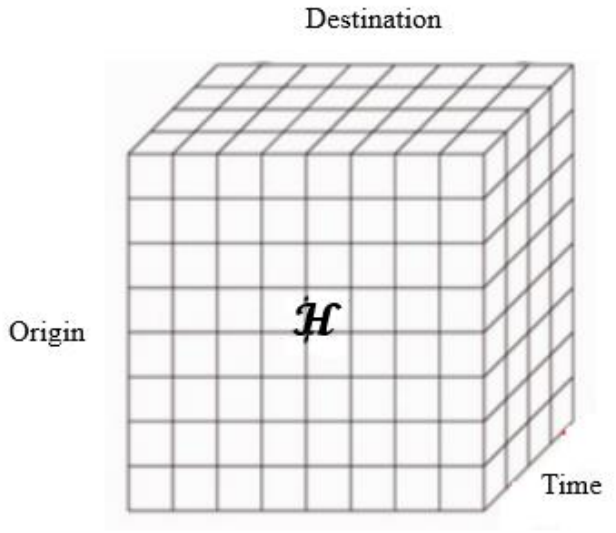


Figure 3.2 Tensor model of human spatial-temporal movements

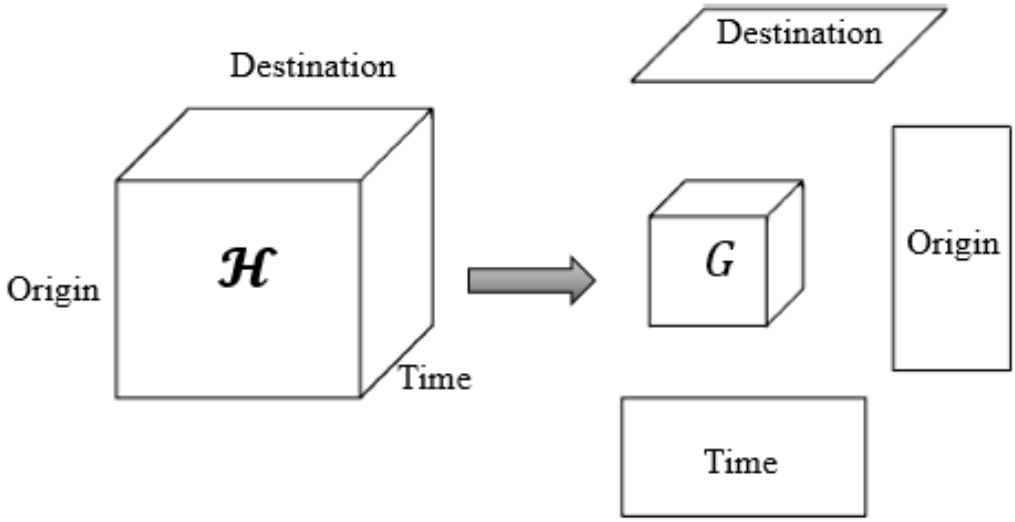


Figure 3.3 Tensor factorization

3.2 Prediction Using Gaussian Process Regression (GPR)

3.2.1 GPR Model between Spatial-Temporal Activities and Latent Features

After the extraction of latent spatial-temporal features, we mathematically model the relationship between spatial-temporal activities such as human mobility and the extracted latent features for prediction. For this, we assume that people's mobility is generated from a smooth and continuous process. This process has typical amplitude and variations in the function which takes place over spatial, temporal, and other characteristics. For

example, to predict the volume of outflow $x_{o_{i,l}}$ in the neighborhood i during time period l (or the volume of inflow $x_{\iota_{i,l}}$), we can model the relationship as below:

$$x_{o_{i,l}} = g(\mathcal{S}_{o_{i,:}}, \mathcal{T}_{l,:}, x_{o_{i,l-1}}, \dots) \qquad (3.4)$$

$$x_{\iota_{i,l}} = g(\mathcal{S}_{d_{i,:}}, \mathcal{T}_{l,:}, x_{\iota_{i,l-1}}, \dots) \qquad (3.5)$$

Note that instead of relating this relationship to some specific models such as linear, quadratic, cubic, or even non-polynomial models, which may have numerous possibilities, we modeled this relationship as a free-form Gaussian process. One reason for using the Gaussian process is that for any spatial-temporal activity y (e.g., $x_{o_{i,l}}$) to be predicted, it will likely be generated by the same process and have similar values as the historical processes that share similar latent spatial-temporal features. We can take advantage of this relationship and use it for prediction. Formally, the Gaussian process can be represented as (Rasmussen 2006):

$$\vec{y} \sim g(\mathbb{X}) \sim GP\left(m(\mathbb{X}), K(\mathbb{X}, \mathbb{X})\right) \qquad (3.6)$$

where $\vec{y}$ is a vector that contains a series of spatial-temporal activities $(y_1, y_2, \dots, y_n)$, $\mathbb{X}$ is the features matrix of $\vec{y}$ (here for an activity $x_{o_{i,l}}$, the corresponding feature in $\mathbb{X}$ would be $(\mathcal{S}_{o_i}, \mathcal{T}_l, x_{o_{i,l-1}}, \dots)$); $m(\mathbb{X})$ is the expected value of the generating process $g(\mathbb{X})$; and $K(\mathbb{X}, \mathbb{X})$ is the covariance matrix where its element $k_{i,j}$ measures the similarity between the input features of activity y_i and y_j. We can also represent the relationship above as:

$$p(\boldsymbol{y}(\mathbb{X})) \sim \mathcal{N}(m(\mathbb{X}), K(\mathbb{X}, \mathbb{X})) \qquad (3.7)$$

For a future activity y^* to be predicted, we have:

$$p\begin{pmatrix}\vec{y}\\ y^*\end{pmatrix} \sim \mathcal{N}(\begin{pmatrix}m(\mathbb{X})\\ m(\mathbb{X}^*)\end{pmatrix}, \begin{bmatrix}K & K^{*T}\\ K^* & K^{**}\end{bmatrix}) \qquad (3.8)$$

where K, K^*, and K^{**} are the abbreviations of the covariance matrix $K(\mathbb{X}, \mathbb{X})$, $K(\mathbb{X}^*, \mathbb{X})$, and $K(\mathbb{X}^*, \mathbb{X}^*)$, respectively, and T indicates a matrix transposition. The key ideas in Equation-3.7 and Equation-3.8 are that we assume that future data are generated from the same process as the existing data. In other words, the future data and existing data have the same distribution. This is a reasonable assumption, since the characteristic of many spatial environments and temporal periods, as well as the patterns of corresponding spatial-temporal activities, are usually stable and will not change significantly over a short period of time.

Since we already have historical datasets, we are more interested in the conditional probability of $p(y^*|\vec{y})$ that given the exiting datasets, what is the probability distribution of an unknown value y^*. Based on the transformations given by Rasmussen (Rasmussen 2006), this conditional probability distribution is:

$$y^*|\vec{y} \sim \mathcal{N}(m(\mathbb{X}^*) + K^*K^{-1}(\vec{y} - m(\mathbb{X})),\ K^{**} - K^*K^{-1}K^{*T}) \qquad (3.9)$$

The best estimate for y^* is the mean value of this distribution:

$$y^* = m(\mathbb{X}^*) + K^*K^{-1}(\vec{y} - m(\mathbb{X})) \qquad (3.10)$$

3.2.2 Prediction of the Volume of Outflow/Inflow

Based on the inference above, in our problem, the prediction for the volume of outflow $x_{o_{i,l}}$ became (similar for $x_{\iota_{i,l}}$):

$$x_{o_{i,l}} = m(\mathbb{X}^*) + K^* K^{-1}(\overrightarrow{x_o} - m(\mathbb{X}^*)) \qquad (3.11)$$

Many applications generally assume that the mean function $m(\mathbb{X})$ is a constant value, e.g., 0. Here we assume $m(\mathbb{X})$ is a constant C_o .

$$x_{o_{i,l}} = \mathsf{C}_o + K^* K^{-1}(\overrightarrow{x_o} - \mathsf{C}_o) \qquad (3.12)$$

Note that in the input features, we have past values $x_{o_{i,l-1}}$, ...; here, we only consider one step backwards $x_{o_{i,l-1}}$.

One problem is that the input feature ($\mathcal{S}_{o_{i,:}}$, $\mathcal{T}_{l,:}$, $x_{o_{i,l-1}}$) of $x_{o_{i,l}}$ contains three variables, the spatial latent feature $\mathcal{S}_{o_{i,:}}$, the temporal latent feature $\mathcal{T}_{l,:}$, and the past outflow volume $x_{o_{i,l-1}}$, each having different meanings, amplitudes, and dimensions. To collectively consider the spatial factors, temporal factors, and flow volume, we design a new covariance function:

$$k\left(\left(\mathcal{S}_{o_{i_1,:}}, \mathcal{T}_{l_1,:}, x_{o_{i_1,l_1-1}}\right), \left(\mathcal{S}_{o_{i_2,:}}, \mathcal{T}_{l_2,:}, x_{o_{i_2,l_2-1}}\right)\right) = \sigma_s^2 \exp\left(-\frac{1}{2l_s^2}|\mathcal{S}_{o_{i_1,:}} - \mathcal{S}_{o_{i_2,:}}|^2\right) + \sigma_t^2 \exp\left(-\frac{1}{2l_t^2}\left|\mathcal{T}_{l_1,:} - \mathcal{T}_{l_2,:}\right|^2\right) + \sigma_p^2 \exp\left(-\frac{1}{2l_p^2}|\, x_{o_{i_1,l_1-1}} - \, x_{o_{i_2,l_2-1}}|^2\right) \qquad (3.13)$$

where σ_s, σ_t, σ_p, l_s, l_t, l_p are all hyper parameters to be inferred, while $|\mathcal{S}_{o_{i_1,:}} - \mathcal{S}_{o_{i_2,:}}|$, $\left|\mathcal{T}_{l_1,:} - \mathcal{T}_{l_2,:}\right|$, and $|\, x_{o_{i_1,l_1-1}} - \, x_{o_{i_2,l_2-1}}|$ are the Euclidean distance between latent

spatial features, temporal features, and past outflows, respectively. Equation 3.13 computes the differences between spatial features, temporal features, and mobility in isolated infinity dimensional spaces and merges them. Therefore, by defining the covariance function like this, the predictions made through Equation 3.12 are based on the historical datasets of different (but similar) spatial areas, temporal time periods, and mobility trends, instead of just one specific neighborhood and time period of interest.

3.2.3 Flow between Neighborhoods

With the predicted outflow (inflow) of each neighborhood, we could further predict the flow between any two neighborhoods. One problem here is that the flow between any two neighborhoods could be relatively sparse and has unstable temporal pattern, which makes it difficult to model and predict directly. However, based on our observations, for a given neighborhood, the ratio of trips heading to different neighborhoods during a specific time period is relatively stable. So we propose to predict $\vec{\theta}_{i,l} = (\theta_{i,l,1}, \dots \theta_{i,l,j}, \dots)$ first, where $\theta_{i,l,j}$ is the percentage of vehicles which start from neighborhood i would head to neighborhood j during time period l as:

$$\vec{\theta}_{i,l} = \beta \times \hat{\theta}_{i,l} + (1 - \beta) \times \vec{\theta}_{i,l-1} \qquad (3.14)$$

$$\sum_j \theta_{i,l,j} = 1 \qquad (3.15)$$

Where β is a constant parameters between 0 and 1, and $\hat{\theta}_{i,l}$ is the corresponding history average value of $\vec{\theta}_{i,l}$. Intuitively, this equations uses a weighted sum model to predict $\vec{\theta}_{i,l}$ based on the corresponding values of its history and previous hour.

Lastly, with $x_{o\,i,l}$ and $\theta_{i,l,j}$, we can compute $x_{i,l,j}$, the number of trips starting from neighborhood i heading to neighborhood j during time period l as:

$$x_{i,l,j} = x_{o\,i,l} \times \theta_{i,l,j} \quad (3.16)$$

4.0 TRAJECTORY DISTRIBUTIONS IN THE ROAD NETWORK

After predicting the flow between neighborhoods, this section further presents how we modeled and estimated the corresponding trajectory distributions in the road network, based on the previously predicted flow volume. We first give the mathematical definition of trajectory distributions. The simulation of the trajectory distributions comprises two parts: (1) predicting the flow volume between the origin and destination road segments; and (2) finding the probable trajectories between the origin and destination road segments and estimating their corresponding possibilities. We will describe how to solve these two sub-problems in detail.

4.1 Definitions

We will first provide the symbols and definitions of road network, trajectory, and trajectory distributions respectively.

The road network can usually be viewed as a directed graph $G = (V, E)$, where E represents the set of road segments and V is the set of vertices that represent the road's end points or the intersections between road segments.

Trajectory tr can be thought of as a series of consecutive road segments with location information that a vehicle/person passes by. In particular, we define $tr = (e_{i1}, e_{i2}, .., e_{im})$, where e_i is a road segment in the road network.

In this thesis, we are more interested in the eventual traffic situation. So instead of studying the trajectory of an individual user, we focus on the overall distribution of trajectories throughout a city level's road network. Mathematically, we define the trajectory distribution as $trd = ((e_{i1}, e_{i2}, .., e_{im}), \mu)$, where $(e_{i1}, e_{i2}, .., e_{im})$ is a trajectory, while μ is the estimated number of people or vehicles that would follow this trajectory. Figure 4.1 gives an example of trajectory distribution $trd = \big((e_1, e_2, e_3, e_4, e_5, e_6), 3\big)$, which indicates that there are three vehicles that would follow the trajectory $(e_1, e_2, e_3, e_4, e_5, e_6)$.

To infer all the trajectory distributions in the road network, there are two specific questions that must be answered:

(1) Given any pair of origin and destination road segment (e.g., e_1 and e_2), how many vehicles will travel from segment to another?

(2) What are the probable trajectories that people would follow from the origin road segment to the destination road segment, and what is the corresponding possibility of each trajectory?

We will address these two questions in the next subsections including their challenges, and our proposed solutions.

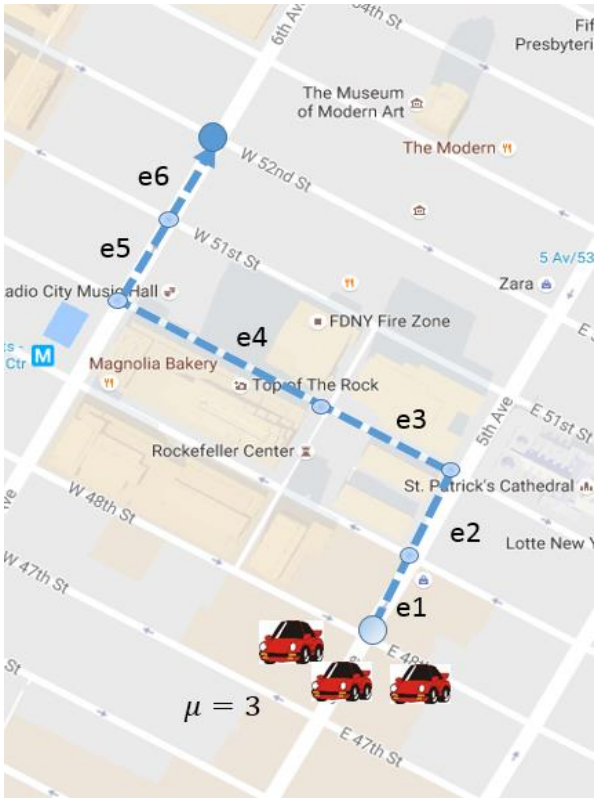


Figure 4.1: An illustration of a trajectory distribution

4.2 Flow Volume Between Road Segments

The traffic that moves from one road segment to another over a short time period could be sparse, which would make it difficult to directly predict. Because we are more interested in the overall traffic situation in a city level, we could take advantage of the previously predicted flow of traffic between any two neighborhoods. Based on these

predictions, we could further estimate the corresponding flow volume between any two road segments.

In particular, a trip that would head from one neighborhood (e.g., neighborhood i) to another neighborhood (e.g., neighborhood j), it could start from any road segment in neighborhood i and end in any road segment in neighborhood j. But in the real world, we might find that some road segments are more popular as origins and some road segments are more popular as destinations during different time periods. For example, a road segment in New York City that includes a large office building such as One World Trade Center, the tallest building in New York with 104 stories and 3 million square feet of office space (WorldTradeCenter 2017), would definitely be a much more popular destination in the morning and origin in the evening, respectively, as compared with other road segments. Given the number of people/vehicles heading from neighborhood i to neighborhood j, in order to estimate how likely they would start from a road segment $\dot{\imath}$ (in origin neighborhood i) and end at another road segment $\dot{\jmath}$ (in destination neighborhood j), we adapt the idea of a spatial interaction gravity model, as proposed by (Wilson 1967). We first estimate the spatial interaction level between any origin road segment $\dot{\imath}$ (in neighborhood i) and destination road segment $\dot{\jmath}$ (in neighborhood j) during time period l as:

$$f(\dot{\imath}, \dot{\jmath}, l) = \mathcal{G}\frac{w_{o_{\dot{\imath},l}} \times w_{\iota_{\dot{\jmath},l}}}{d_{\dot{\imath},\dot{\jmath}}} \qquad (4.1)$$

where $\mathcal{G}$ is a constant parameter, $w_{o_{i,l}}$ is the weight of road segment i as the origin during time period l, $w_{\iota_{j,l}}$ is the corresponding weight of road segment j as the destination, and $d_{i,j}$ is the Euclidean distance between them. It is worth noting that some previous works use different categories of data to approximate the weight w. Among all those categories of data, one of the most widely used is the population of corresponding spatial area (Hua and Porell 1979)-but the static population of corresponding area does not work in this scenario. One major reason is that because we focus on the short term prediction, e.g., a city level's mobility in an hour, while the population feature might be more suitable for some long-term and static prediction. For example, in urban areas, especially those central business districts, people come and go from time to time every day, making it impossible to accurately count or even estimate the population of each area every hour. As a result, we would like to estimate weight w based on our history mobility dataset. In particular, in our implementation, we use the historical average number of trips that started from road segment i during time period l as the weight $w_{o_{i,l}}$, and the corresponding historical average number of trips that ended at e_j as the weight $w_{\iota_{j,l}}$.

Instead of estimating a constant value for $\mathcal{G}$ like some previous works, we propose to normalize the interaction level between each pair of road segments i and j in origin neighborhood i and destination neighborhood j, and multiply it by $x_{i,l,j}$ (the flow volume from neighborhood i to neighborhood j), in order to obtain the flow volume between

those road segments. Eventually, $xe_{i,l,j}$, the number of vehicles that are heading from road segment i (in neighborhood i) to road segment j (in neighborhood j) during time period l is computed as:

$$xe_{i,l,j} = x_{i,l,j} \frac{\frac{wo_{i,l} \times w\iota_{j,l}}{d_{i,j}}}{\sum_p \sum_q \frac{wo_{p,l} \times w\iota_{q,l}}{d_{p,q}}} \quad (4.2)$$

The intuition behind this equation is that if the road segments e_i and e_j have strong spatial interaction during time period l given the historical dataset, a new trip heading from neighborhood i to neighborhood j will also be likely to start from road segment e_i and end at e_j then.

4.3 Trajectory Distribution Simulation

After the estimation of flow between road segments in the road work, we turn to our second question: What are the probable trajectories of vehicles heading from one road segment to another and the corresponding possibility of each trajectory?. This problem is also nontrivial, due to the fact that there are usually multiple routes for a vehicle to travel from one place to another in the road network. Figure 4.2 shows an example of the different types of trajectories that can be used to travel from one road segment to another.

There are different strategies we can use to infer a trajectory. For example, we can observe user driving patterns (such as how likely they are to make a right turn at a specific intersection) from historical trajectories (Liu and Karimi 2006, Froehlich and Krumm 2008, Jeung, Yiu et al. 2010). However, these strategies require users to keep uploading their GPS points frequently, sometimes as often as every two minutes, which can be difficult to acquire, due to both privacy and technical issues. Besides, many people will simply follow the directions of Google Maps or Waze when they are heading to some places, and as a result, there is no personal routing preference, as some of these studies claim.

In this paper, we propose different general models to estimate trajectories and simulate the corresponding trajectory distributions, instead of focusing on the exact trajectory of each individual user. One simple trajectory simulation model is to use the shortest path between any two places, as done by some previous works (Matthias and Zuefle 2008, Deri, Franchetti et al. 2016). Mathematically, assuming that the shortest path between road segment e_i and e_j is $tr_{i,j}^1$, then the possibility that vehicles that are heading from e_i to e_j would follow $tr_{i,j}^1$ is:

$$h\left(tr_{i,j}^1\right) = 1 \quad (4.3)$$

The corresponding trajectory distribution would be:

$$trd_{i,j}^1 = \left(tr_{i,j}^1,\ xe_{i,l,j} * 1\right) \quad (4.4)$$

However, in practice, while people will not always follow the shortest path from one place to another, they are also unlikely to make long detours. Based on this observation, we propose the following two trajectory distribution simulation methods.

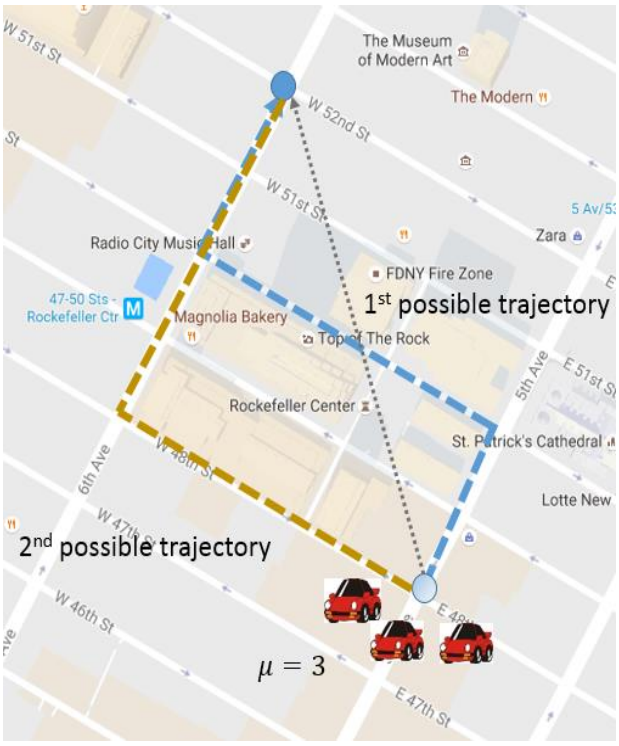


Figure 4.2 Some possible trajectories for a given origin-destination pair.

The first simulation method is that to go from one place to another, we assume people would take one of the top-K shortest paths with equal probability. Mathematically, assuming that $tr_{i,j}^{k}$ is one of the top-K shortest paths between road segment e_i and e_j,

then the possibility that vehicles that are heading from road segment e_i to e_j would follow $tr_{i,j}^k$ is:

$$h\left(tr_{i,j}^k\right) = \frac{1}{K} \quad (4.5)$$

The corresponding trajectory distribution:

$$trd_{i,j}^k = \left(tr_{i,j}^k,\ xe_{i,l,j}/K\right) \quad (4.6)$$

Taking one of the top-K shortest paths might more accurately portray people's daily driving behaviors rather than assuming that they always follow the shortest path. However, due to the complexity of the road network's structure, people's driving preference might be skewed rather than equally prefer any one of the top-K shortest paths. For example, taking the $k+1^{th}$ shortest path sometimes might result in much more extra travel distance compared with the k^{th} shortest path, and as a result, people will be careful to avoid that particular path. Instead of assuming that people would take any one of the top-K shortest paths with equal probability, we would estimate the probability of each trajectory, based on their actual distance and the distance of theoretical shortest path, given the historical dataset. For example, given a pair of origin and destination road segments whose shortest travel distance is 10 miles, what is the probability that people would take a path with the distance of 11.5 miles, 12 miles, 15 miles, or 20 miles? Although it is difficult to collect detailed GPS points from every anonymous trip, we could know the miles of each trip through the odometer, which is a common feature of all vehicles. Consequently, we estimate the possibility of each trajectory by its actual

distance and theoretical shortest path's distance through a multivariate kernel density estimation (Simonoff 1996). Formally, for vehicles heading from road segment e_i to e_j, the possibility of following a trajectory $tr_{i,j} = (e_i, .., e_j)$ is:

$$h(tr_{i,j}) = \frac{1}{n}\sum_c (2\pi)^{-1} |H_{i,j}|^{-\frac{1}{2}} \mathcal{K}(\vec{z} - \vec{z}_c) \quad (4.7)$$

$$\mathcal{K}(\vec{z}) = e^{-\frac{1}{2}\vec{z}^T H_{i,j}^{-1} \vec{z}}, \quad (4.8)$$

$$\vec{z} = (|tr_{i,j}^1|, |tr_{i,j}| - |tr_{i,j}^1|), \quad (4.9)$$

where $\mathcal{K}()$ is the kernel function, $\vec{z}_c$ is a history record, $tr_{i,j}^1$ indicates the shortest path from road segments i to j, and H is the bandwidth matrix (covariance matrix). It is worth noting that in order to increase the estimation accuracy of trajectory possibilities (equation 4.7), we compute the bandwidth matrix $H_{i,j}$ for each pair of origin neighborhoods i and destination neighborhoods j, instead of using the same bandwidth matrix H for all the trips. The major reason for doing this is that the road network structure between different pairs of origin and destination neighborhoods could be very different, which makes people's driving preferences and the corresponding trajectory distributions vary. As a result, the parameters (the bandwidth matrix**)** between each pair of origin and destination neighborhoods should also vary.

Based on this possibility, we propose a top-K likely trajectory distribution simulation strategy that for any given pair of origin and destination road segments, we would find the trajectories that have one of the top-K largest possibilities based on the historical dataset. Mathematically, we model the problem as:

$$trd_{i,j}=\left(tr_{i,j},\ xe_{i,l,j} \times \tilde{h}\left(tr_{i,j}\right)\right) \qquad (4.10)$$

$$\hat{h}(tr_{i,j}) = h(tr_{i,j}) / \sum_{tr_{p,q}} h(tr_{p,q}) \qquad (4.11)$$

where $tr_{i,j}$ is a trajectory from road segment i to j with one of the K largest possibilities $h(tr_{i,j})$.

Note that we would keep the trajectory simulation as an independent module. By doing so, people can also try other trajectory simulation methods besides the proposed methods here and use the one that is most suitable for their application. For example, when there are a certain amount of self-driving vehicles in the road network, the prediction system can simulate those self-driving vehicles' trajectories through adapting their routing strategy, such as taking one of the fastest paths by aggregating the collected traffic information.

4.4 Trajectory Distributions Analysis and Applications

After the simulation of the trajectory distributions, we can further process and analyze the synthetic data for a great deal of interesting information, such as predicting hot road segments with high centrality where many vehicles would pass by, which might be an indication of potential traffic jams or bottlenecks. We could simply go over each

trajectory distribution, sum the number of people/vehicles that would pass through the specific road segments, and output those hot road segments. It is worth pointing out that there could be different definitions of hot road segments under different scenarios, such as the road segments with the top-K largest traffic volumes, or the road segments that have a traffic volume that is larger than a given threshold. Our methodology is flexible and can handle either definition, but to be consistent in this paper, we adopted the first definition and will output the hot road segments with the top-K largest traffic volume later in the experiment.

Besides the prediction of hot road segments where potential traffic jams might form, we are able to further predict and reveal the formation of them; namely, what are the top-K primary origin/destination neighborhoods of the traffic that is passing through those hot road segments? This is a major advantage of our methodology as compared with traditional traffic prediction, which focuses on predicting an individual road segment's traffic situation but provides little additional information about the origins or destinations of those vehicles, which is a vital element for understanding the formation of some traffic jams.

5.0 LARGE-SCALE TRAJECTORY DISTRIBUTION SIMULATION

The problem of trajectory distribution simulation is computationally intensive and difficult to accomplish under real-time constraints, because the scale of a metropolitan city's road network and the corresponding number of trajectories that people might choose to take during a certain time period could both be extremely large. To tackle this challenge, we present a MapReduce-based distributed solution. Based on the synthetic trajectory distributions, we further design different MapReduce-based algorithms to predict the hot road segments and identify the popular origins/destinations of the traffic passing through those hot road segments of interest.

5.1 MapReduce-Based Trajectory Distribution Simulation

To implement the simulation methods from Section 4, one key step is to find the probable trajectories, namely, the top-K shortest paths for each pair of origin and destination road segments. A naive algorithm is to simply enumerate all possible routes between any two road segments, which would cost $O(2^{|E|})$. This is not an acceptable level of performance, especially for real-time decision making, given that the number of road segments E in a

city level could be in the range of tens of thousands. We can improve this time complexity by using Yen's top-K shortest paths algorithm (Yen 1970), which would take $O(K * |E|^2 * \log(|E|))$ to compute each pair's top-K shortest paths, if it is optimized with a priority queue. For all pairs' top-K shortest paths, it would still take $O(K * |E|^4 * \log(|E|))$, which is computationally intensive and requires efficient algorithms for a real-time response.

To tackle this problem, here we propose a MapReduce-based distributed algorithm to simulate all the trajectory distributions in the road network. To be clear, we do specifically give the algorithm of the top-K likely trajectory distribution simulation discussed in the Section 4, but our algorithm is very flexible and can handle all the models of trajectory distribution discussed in the Section 4.

Algorithms 5.1 and 5.2 show the pseudo-code in detail. The general idea is that in the Map phase, we distribute the flow volume xe between each pair of road segments to the reduce phase. The key of the Map phase output is the id of the origin road segment, and the values of the Map phase output are the corresponding destination road segments and flow volumes. In this way, the fluxes between each pair of road segments will be aggregated in the Reduce phase, based on the origin road segments. As previously discussed, the weights of different road segments are unevenly distributed. Some road segments might have almost zero people either starting or ending there during certain time periods. To reduce the amount of data to be processed and increase the time performance of the program, we could skip some of the trips that few people took in the

past. Each Reduce task will be in charge of searching the trajectories with increasing distance that start from the given road segment, namely, e_i. For each found probable trajectory, we compute its corresponding possibility and flow volume, then output it.

Since the map stage (Algorithm 5.1) is pretty straightforward and the reduce stage (Algorithm 5.2) is the core of our trajectory distribution simulation, we will go over it in detail. During the description of the algorithm, we use the word "path" and "trajectory" interchangeably, since they both indicate a series of road segments. In lines 1–3, we read in the processed data, such as the road network, bandwidth matrices H, and the history trip records tr_c from disk (the Hadoop distributed file system). In line 4, we initialize an array s, where s_j would store the length of the shortest path from origin road segment e_i to e_j. With the help of array s, we can skip the trajectories that are long detours for the given threshold (line 14) and improve the performance of our algorithm. In line 5, we construct a min heap Q to store the destination road segments and the corresponding distances (from origin road segment e_i to them) for a trajectory search. With such a min heap Q, we can get and update the smallest record with only $O(1)$ and $O(\log(N))$ time, respectively. In line 6, we use an array of min heap R_j to keep track of the trajectories with the top-K highest possibilities ending at road segment e_j. In line 7, we store each node's parent node in order to rebuild the corresponding trajectory. Note that since we are interested in finding several probable trajectories between each pair of origin and destination road segments (rather than a single shortest path), we need to keep track of all

the corresponding parent nodes, based on the distance. For example, if there is a path from e_i to e_j with a total length of d, we store the previous road segment of e_j as $parent_{j,d}$. In other words, there is a path from from e_i to e_j, $<e_i,\ldots, parent_{j,d}, e_j>$, which has a total length of $d + |e_j|$. Within the while loop that starts from line 8 to line 35, we process the path, starting from the origin road segment, with increasing distance. During each iteration, when we have a path ending at road segment e_j, we check that if the path is a long detour by comparing it to the theoretical shortest path (line 14). If it is a long detour, we skip the path since people are unlikely to take long detours during the course of their daily driving. Otherwise, we proceed with processing the trajectory. To save storage space, we only store the last road segment of each path during the search, and rebuild the whole trajectory through iterating the parent pointers (lines 16–19). In line 21, without a loss of generality, we compute the possibility of the trajectory with a multivariate kernel density estimation (Equation 4.10). After we finish processing the current found trajectory, we expand the search and update the adjacent road segments of the finalized road segment (e_j) and push the updated values into the min heap Q (lines 28–33). Note that during people's daily driving, they seldom pass the same road segment multiple times in a trip (unless they get lost or find themselves in other uncommon situations). As a result, during the search, we only update the adjacent road segments that have not yet been visited by the current trajectory in order to avoid duplicate road segments (line 30). Finally, we compute the volume of vehicles that would follow the found trajectory and output the corresponding trajectory distributions (lines 36–41).

We will also provide the time complexity analysis of Algorithm 5.2. First, let's assume that, based on the *threshold* we set in line 14, each road segment will be visited a maximum of U times, so the while loop (line 8–line 35) will be executed a maximum of UE times. Within the while loop, there are several major operations. The first operation is to find the destination road segment with current minimal distance (line 9–10). Since we use the min heap, the time complexity of this operation is $\log(UE)$. The second operation is reconstructing the whole trajectory, based on the parent pointers (lines 16–19), which will be executed a maximum of $O(E)$ times. The third operation is to compute the possibility of the found trajectory, based on the history records in line 21 (assume that there are M records). If necessary, we then update the min heap R_j with time complexity of O(logK) (lines 22–27). The last operation is to update the adjacent road segments (lines 28–33). Note that in the road network, the degree of each road segment is relatively stable and small. For example, most road segments would have a maximum of three to four adjacent road segments. Hence, updating the adjacent road segments and checking the duplicate road segments would simply cost $O(E)$ time. When considering all the factors, the overall time complexity of Algorithm 5.2 is $O(UE * (\log(UE) + \log(K) + E + M))$. For the simulation, we need to compute the trajectory distributions starting from all the road segments, and we assume that there are $\mathcal{R}$ reducers available in the Hadoop cluster. The final time complexity of the MapReduce based trajectory distribution simulation is $O(\frac{UE^2 * (\log(UE) + \log(K) + E + M)}{\mathcal{R}})$.

Algorithm 1 Traj_Dist_Sim_Map

Input: The flow volume $x_{I,J}$ from neighborhood I to J during time period l

Output: The flow volume xe for road segment i in neighborhood I to all road segments j in neighborhood J

```
Read the road network G = {V, E} from disk
Read the average trip numbers w_o and w_ι from disk.
Get neighborhood I and J from G
weight_sum = Σ_i Σ_j w_{o_{i,l}} * w_{ι_{j,l}} / d_{i,j}   (w_{o_{i,l}} * w_{ι_{j,l}} / d_{i,j} > threshold)
for each road segment i in neighborhood I do
   for each road segment j in neighborhood J do
      if w_{o_{i,l}} * w_{ι_{j,l}} / d_{i,j} >= threshold then
         Emit the key-value pair (e_i, {e_j, x_{I,J} * (w_{o_{i,l}} * w_{ι_{j,l}} / d_{i,j}) / weight_sum})
      end if
   end for
end for
```

Line 4 in LaTeX: $weight\_sum = \sum_i \sum_j w_{o_{i,l}} * w_{\iota_{j,l}} / d_{i,j}$, where the sum is taken over $w_{o_{i,l}} * w_{\iota_{j,l}} / d_{i,j} > threshold$

Line 8 in LaTeX: $(e_i, \{e_j, x_{I,J} * \frac{w_{o_{i,l}} * w_{\iota_{j,l}} / d_{i,j}}{weight\_sum}\})$

Algorithm 5.1. Map phase of trajectory distribution simulation.

Algorithm 2 Traj_Dist_Sim_Reduce

Input: key:the origin road segment e_i, value:a list of destination road segments e_j and the corresponding flow volume $xe_{i,l,j}$

Output: a list of top-K likely trajectory distributions $trd_{i,j}$ started from e_i

Read in the road network $G = \{V, E\}$
Read in the bandwidth matrices $H_{I,J}$ for each neighborhood pair
Read in all history distance distribution records $\{(||tr^1_{i,j}||, ||tr_c|| - ||tr^1_{i,j}||)\}$
Initialize all $s_j = \{\}$
$Q = Q \cup \{(e_i, ||e_i||)\}$ // Initialize min heap for path searching
$R_j = \{\}$ // Initialize min heap for recording maximal likely paths
$parent_{i,||e_i||} = (-1, 0)$ // Initialize the parent pointer
while $Q \neq empty$ **do**
  $(e_j, d) = min_d\{(e, d)|(e, d) \in Q\}$
  $Q = Q - (e_j, d)$
  if s_j is empty **then**
    $s_j = d$
  end if
  if $d/s_j \leq threshold$ **then**
    $tr = \{\}$, $d_x = d$, $e_x = e_j$
    while $d_x \neq 0$ **do**
      $tr = (e_x, tr)$
      $(e_x, d_x) = parent_{x,d_x}$
    end while
    Locate tr's origin and destination neighborhood I and J
    $h = \frac{1}{n}\sum(2\pi)^{-1}|H_{I,J}|^{-1/2}\mathcal{K}(\overrightarrow{z} - \overrightarrow{z_c})$
    if R_j is empty or $h > min_h\{R_j\}$ **then**
      $R_j = (tr, h) \cup R_j$
    end if
    if $|R_j| > K$ **then**
      Delete the element with smallest h from R_j
    end if
    for each e_x adjacent to e_j **do**
      if $e_x \notin tr$ **then**
        $parent_{x,(d+||e_x||)} = (e_j, d)$
        $Q = Q \cup \{(e_x, d + ||e_x||)\}$
      end if
    end for
  end if
end while
for each R_j **do**
  $sum_j = \sum_{(tr,h)\in R_j} h$
  for each $(tr, h) \in R_j$ **do**
    Output the trajectory distribution $\{tr, xe_{i,l,j} \times h \div sum_j\}$
  end for
end for

Algorithm 5.2. Reduce phase of trajectory distribution simulation.

5.2 MapReduce-based Trajectory Distribution Analysis

Based on the simulation of trajectory distributions, we can predict the hot road segments that have a high degree of centrality, which are likely places for potential traffic jams or bottlenecks to happen. Besides that, we can further identify the primary origin/destination neighborhoods of the hot road segments of interest, from which it would be possible to reveal the causes of potential traffic jams, such as the primary origins and destinations of the traffic in some specific road segments. One major challenge here is that there could be up to $O(KE^2)$ trajectory distributions outputted from the previous simulation step. Considering that there are tens of thousands of road segments in a city level's road network (and especially in a major metropolitan area), there could be almost one billion generated trajectory distributions. As a result, MapReduce-based distributed algorithms are specifically designed for the analysis of trajectory distributions.

For the hot road segments, we propose a flow-volume-based dynamic betweenness centrality to measure the popularity of each road segment during a specific time period in the sub-section 4.3. Intuitively each road segment's dynamic betweenness

centrality equals the aggregated number of people/vehicles that would pass it based on our synthetic traffic distributions. Our generated trajectory distributions are a good source to compute such a dynamic betweenness centrality. We could simply go over each trajectory distribution, sum the number of people/vehicle that would pass each specific road segment, and output the hot ones through ranking. The pseudocode of the designed MapReduce based hot road segment prediction is shown in Algorithms 5.3 and 5.4. Generally, we send the synthetic trajectory distributions to different mappers in the Algorithm 5.3. The mappers go over each road segment of the passed-in trajectory and the corresponding traffic volume. Then the reducers will get the id of each road segment as the key, and a list of traffic volume as the values so we can sum them up. After that, we can use a simple sorting algorithm to quickly identify the hot road segments with the top-K highest traffic volume—or the road segments with a traffic volume higher than a given threshold.

Algorithm 3 Hot_Roads_Mining_Map

Input: A trajectory distribution $trd = (tr, u)$

Output: A series of road segments (belonging to the input trajectory) and the corresponding traffic volume.

1: **for** each road segment e_i in the tr **do**
2: Emit the key-value pair (e_i, u)
3: **end for**

Algorithm 5.3. Map phase of Hot Roads Prediction.

Algorithm 4 Hot_Roads_Mining_Reduce

Input: key:a road segment e, value:a list of traffic volume $u_1, u_2, ...$
Output: the road segment and the sum of traffic volume that passes it

1: **for** each value $u_i \in \{u_1, u_2, ...\}$ **do**
2: $sum+ = u_i$
3: **end for**
4: Output the value (e, sum)

Algorithm 5.4. Reduce phase of Hot Roads Prediction.

After predicting those hot road segments, a city agency might also want to further investigate the top-K major origins or destinations of the traffic that passes through one or more specific hot road segments, which is essential to identify the causes of those traffic jams. Such information could also be used to optimize the road network, public transportation systems, and emergency management. For example, if the police want to block several streets for events later in a given day, by querying the major origin/destination neighborhoods where people would pass by at that time, the system could send notifications to corresponding drivers or even to self-driving vehicles so that they could update their schedules or routing. We provide the corresponding MapReduce-based algorithm for these scenarios, as shown in Algorithms 5.5 and 5.6. Intuitively, the algorithms work similarly to Algorithms 5.4 and 5.5. The synthetic trajectory distributions are sent to different mappers, which will go over each road segment. If the road segment is one of those in which we are interested, we pass its origin and destination

neighborhoods and amount of corresponding traffic volume to the reducers, and the reducers will aggregate the results.

Algorithm 5 Orig_Dest_Mining_Map

Input: A trajectory distribution $trd = (tr, u)$
Output: The origin and destination neighborhood and the corresponding traffic volume.

```
1: Read in the road segments S = {e_i1, ...} of interest.
2: Read in the road network G = {V, E}.
3: Locate tr's origin and destination neighborhood I and J.
4: for each road segment e_i ∈ tr do
5:    if e_i ∈ S then
6:       // tr passes the road segments of interest
7:       Emit the key-value pair (Origin:I, u)
8:       Emit the key-value pair (Destination:J, u)
9:       break
10:   end if
11: end for
```

Algorithm 5.5. Map phase of popular origin/destination mining.

Algorithm 6 Orig_Dest_Mining_Reduce

Input: key:a origin or destination neighborhood N, value:a list of traffic volume $u_1, u_2, ...$
Output: the neighborhood and the sum of traffic volume

```
1: for each value u_i ∈ {u_1, u_2, ...} do
2:    sum+ = u_i
3: end for
4: Output the value (N, sum)
```

Algorithm 5.6. Reduce phase of popular origin/destination mining.

6.0 EXPERIMENT RESULTS

In this section, we present the experimental results of our methodology. In particular, we conducted case studies using the taxi trip data collected from Beijing and New York City. First, we introduce and analyze the collected dataset. Next, we discuss a series of experiments that we conducted to evaluate the accuracy of our methodology, such as (1) the prediction of outflow/inflow across different areas and time periods, (2) the prediction of flow between neighborhoods and (3) the prediction of hot road segments and their primary origin/destination neighborhoods. After that, we investigated the time performance of our proposed MapReduce-based algorithms, particularly in terms of their scalability.

6.1 Dataset

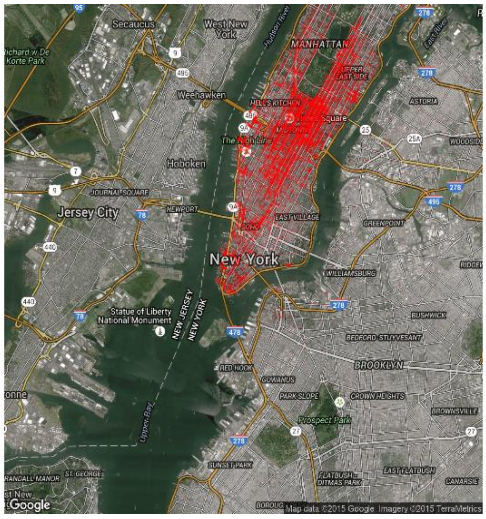
In this thesis, we conduct two cases study through collecting the taxi data from New York City and Beijing. Taxis play a very important transportation role in many metropolitan areas. Given the popularity and the importance of taxis, many previous works view them as the ubiquitous mobile sensors constantly probing a city's rhythm and pulse, such as

traffic flows on road surfaces and citywide travel patterns of people (Zheng, Liu et al. 2011). In New York City, each day almost 13,000 taxis carry over one million passengers and make, on average, 500,000 trips—totaling over 170 million trips a year (Ferreira, Poco et al. 2013). Predicting how people move around through taxis not only help optimize the taxi operation itself, but also reveals the cultural and geographic aspects of the city and detects abnormal events, among other things. It is worth mentioning that our methodology can be applied to diverse mobility datasets (the dataset might contain the detailed trajectories of every trip, or just some origin/destination information), such as census data/results of travel surveys, mobile phone records, check-in data from location-based social networks, and others. In our work, we use the taxi dataset, which could contain detailed trajectories for each trip of the taxi, so that we can compare the results of our trajectory distribution prediction methodology with the ground truth.

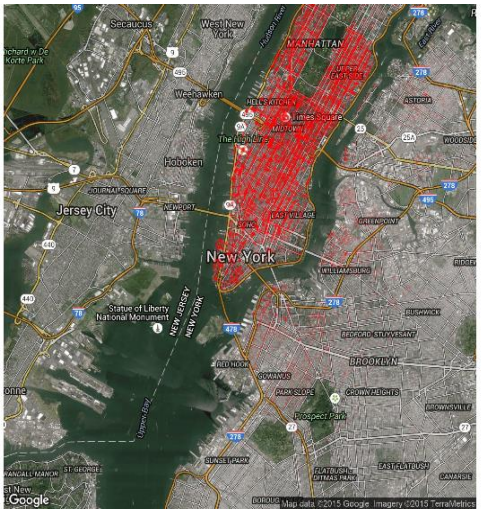
For New York City's taxi trips, we collected data spanning from September 1, 2014, to October 31, 2014, a total of approximately 29 million distinct trip records. The data is shared by the New York government through an open data project named "NYC Open Data" (NYCOpenData 2016) which provides data to the public, including millions of taxi trip records. Each taxi trip record has the pick-up time, pick-up location, drop-off time, drop-off location, and the travel distance, among others. As for Beijing, we obtained the taxi trajectory dataset shared by (Yu, Zhao et al. 2010, Zhang, Zhang et al. 2011). The dataset consists of 27 days of trajectory data recorded from May 1, 2009 to May 29, 2009 (the data from both May 10 and May 20 are missing). The dataset was

collected from 28,000 taxicabs in Beijing, which include approximately 42% of the total number of taxis in Beijing. Compared with the taxi dataset for New York, which only contains the information of origin and destination of each trip, the Beijing taxi dataset contains a series of GPS points uploaded by the taxis every few minutes with additional information (for example, whether the taxi is carrying passengers or not). We divided each taxi's sequentially uploaded points into a series of trips, based on several criteria. The major criterion is that if the status of an uploaded point changes, such as from empty to loaded or vice versa, we will mark the point as the beginning or the end of a trip. Note that the first week of May is a national holiday in China and as a result, people's mobility patterns are quite different from other days; we excluded these days from the experiment.

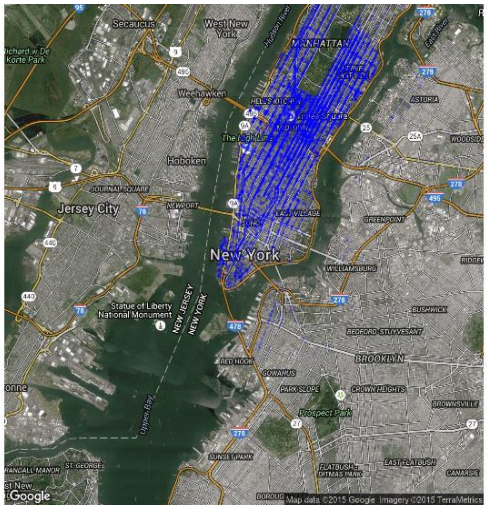
We first visualized NYC's pick-ups and drop-offs distribution in the morning (10:00 – 10:59 am) and at night (09:00 – 9:59 pm) in a randomly selected day in Figure 6.1. From these visualizations we noticed most of the taxi activities happened within the Manhattan district although there were some pick-ups and drop-offs outside the Manhattan at night. Among all the neighborhoods within Manhattan district, the districts near Times Square generally have the most pick-ups and drop-offs. This phenomenon is reasonable since Times Square is a highly commercial district, with many people working there, and a tourist attraction. Another observable interesting phenomenon is that in the lower east district, there are significantly more pick-ups and drop-offs at night compared with the daytime, a sign of night life district. The spatial clustering result in the next subsection based on the extracted latent features will also confirm this.



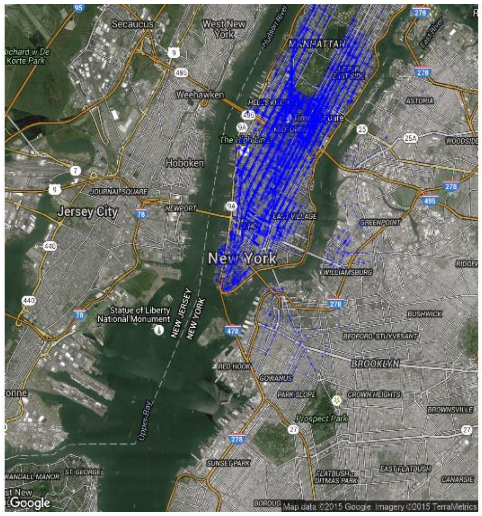
(a) Drop-off activities (10:00-10:59 am)



(b) Drop-off activities (9:00-9:59 pm)



(c) Pick-up activities (10:00-10:59 am)



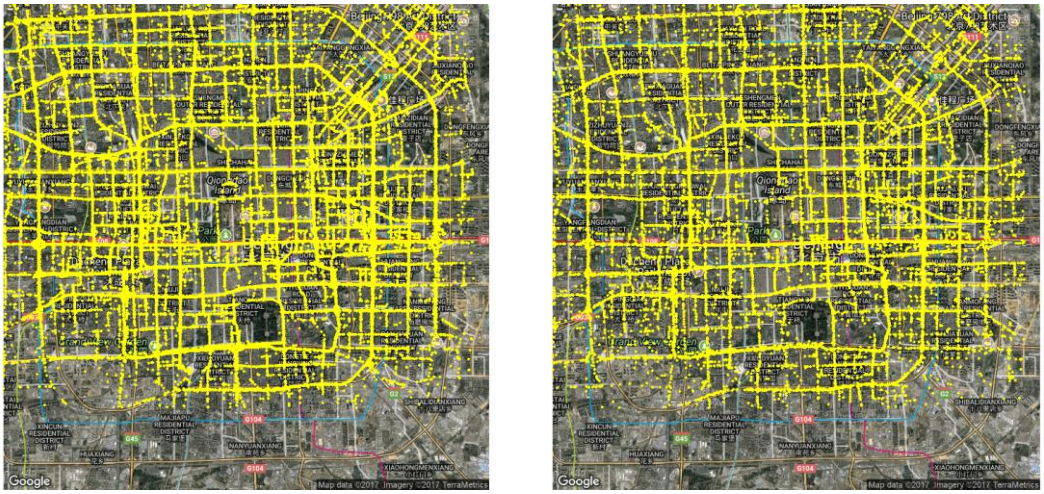
(d) Pick-up activities (9:00-9:59 pm)

Figure 6.1: Pick-up and drop-off activities of NYC in a single day

Since most of the taxi pick-up and drop-off activities happen in Manhattan district, we will focus our analysis on that district. We partitioned the district into small

parallelogram grids, each with approximately 0.8 km on each side. As discussed in (Liu, Liu et al. 2015), while exploring human's spatial-temporal activities with social sensing data, discretizing the studied areas into spatial units with area between 0.25 km^2 and 1 km^2 would be appropriate and has been adopted by many previous works(Reades, Calabrese et al. 2009, Liu, Wang et al. 2012, Toole, Ulm et al. 2012). So the resolution we used (0.64 km^2 per unit) is reasonable and fine enough to demonstrate the accuracy of our prediction methodology in small areas where human's mobility patterns might have high variances.

Besides NYC's data, we also visualized Beijing's taxi activities (the uploaded GPS points) in the morning (10:00 – 10:59 am) and at night (09:00 – 9:59 pm) in a randomly selected day as shown in Figure 6.2. Because the collected taxi data in Beijing is very sparse (containing only 42% of the taxis in Beijing), we partitioned the city into grids with a coarser resolution (with approximately 1.5 km on each side). For both cities, we used one hour as the time unit for the analysis and prediction latter.



(a) Taxi activities (10:00-10:59 am) (b) Taxi activities (9:00-9:59 pm)

Figure 6.2: Taxi activities of Beijing in a single day

6.2 Outflow (inflow) Volume Prediction

With the collected data, we would first investigate the accuracy of our proposed spatio-temporal prediction methodology using the latent features and compared it with existing ones. In particular, for each city we constructed a mobility tensor as described in Chapter 3. Then we conducted the tensor factorization to extract the latent spatial features of each partitioned grid as the origin and destination respectively, and the latent temporal features of each hour. With the extracted latent features, we further trained a Gaussian Process Regression model and used it for prediction. We named our methodology (Gaussian

process regression with latent spatial and temporal features) as GPR-LST for short and compared it with two existing models. One is the parametric seasonal ARIMA model where we take each grid as a fixed point and build seasonal ARIMA models for its time-series outflow and inflow, respectively. Another methodology is the non-parametric model, naive Gaussian Process regression (GPR), which uses the explicit previous time-serious records like ($x_{o_{i,l-1}}, x_{o_{i,l-2}}, x_{o_{i,l-3}},\ldots,$) as the input features and the squared exponential kernel with a separate length scale per predictor as the covariance function. We named this methodology (Naive Gaussian process regression for time series records) as GPR-Naive for short. We have one GPR-Naive model for outflow and one GPR-Naive model for inflow.

We performed all the prediction methodologies on each partitioned grid of the city and predicted each grid's outflow (inflow) in the next hour iteratively. For NYC, we used 4 weeks data as the training dataset and the next 2 weeks data for the verification. For Beijing, we used 8 days data for the training and the rest 3 days for verification. To measure the accuracy of prediction, we used three metrics: (1) root mean squared error (RMSE), (2) mean absolute scaled error (MASE) (proposed by (Franses 2016)) and (3) our designed mean error ratio (MAE). Equation 6.3 – 6.5 show how three metrics are calculated.

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\widehat{y_t} - y_t)^2} \quad (6.3)$$

$$MASE = \frac{\frac{1}{T}\sum_t |\widehat{y_t} - y_t|}{\frac{1}{T-1}\sum_{t=2}^{T} |y_t - y_{t-1}|} \quad (6.4)$$

$$MER = \frac{\sum_{t=1}^{T} |\widehat{y_t} - y_t|}{\sum_{t=1}^{T} y_t} \quad (6.5)$$

Where $\widehat{y_t}$ is the predicted value at time t while y_t is the corresponding ground truth. Note that the general idea of MASE is to compare the prediction methodology with the naive one-step forecast methodology that makes predictions based on the previous value, e.g., to predict human's outflow $x_{o_{i,l}}$ at time period l; the one-step forecast methodology uses the value of $x_{o_{i,l-1}}$ directly. And as for the mean error ratio (MER), we designed it in order to measure the scale of the prediction error vs the ground truth.

We conducted a series of experiments to verify our prediction methodology. We used the prediction error of NYC's outflow in the workday as the baseline, and would like to see how different methodologies perform under different scenarios such as (1) outflow vs inflow, (2) workdays vs weekends, and (3) NYC vs Beijing.

Table 1: Outflow vs Inflow (NYC's Workdays)

	Outflow			Inflow		
	RMSE	MASE	MER	RMSE	MASE	MER
GPR-LST	33.175	0.481	0.096	30.872	0.485	0.097
Seasonal-ARIMA	45.384	0.678	0.133	35.715	0.583	0.115

GPR-Naive	71.865	0.909	0.185	69.575	0.974	0.200

Table 2: Workdays vs Weekends (NYC's outflow)

	Workday			Weekend		
	RMSE	MASE	MER	RMSE	MASE	MER
GPR-LST	33.175	0.481	0.096	32.203	0.655	0.111
Seasonal-ARIMA	45.384	0.678	0.133	42.813	0.880	0.149
GPR-Naive	71.865	0.909	0.185	48.567	0.890	0.151

From the table-1 we can see different methodologies have similar prediction errors when predicting the outflow and inflow. And based on the table-2, it seems several methodologies achieved higher prediction accuracy (made smaller prediction errors) in the workday, which might indicate people's mobility pattern is more regular in the workdays compared with the pattern in the weekends. Generally, from these two tables we can see that our proposed prediction methodology using the latent features achieves the highest accuracy (makes least prediction errors).

We would also like to see how our methodology performs across different cities. So we predicted the outflow of NYC and Beijing in the workdays and the results are shown in table-3. From the table we can see for Beijing, all methodologies achieved less RMSE but had larger MASE and MER compared with NYC. One reason is that the collected taxi data from Beijing is just a small sample of all the taxis (42%) and hence much sparser than the data from NYC. So the average number of taxi activities (pickups and dropoffs) in each partitioned grid of Beijing has a smaller scale than the corresponding one of NYC, resulting smaller RMSE. On the other hand, the sparsity of the data makes the temporal pattern relatively unstable and more difficult to model, resulting in larger MASE and MER. What's more, we have limited data of Beijing's taxi data for training which could all increase the prediction error (MASE and MER). But still, our proposed methodology performs best and achieves least prediction errors among all the methodologies.

Table 3: NYC vs Beijing (Outflow in the workdays)

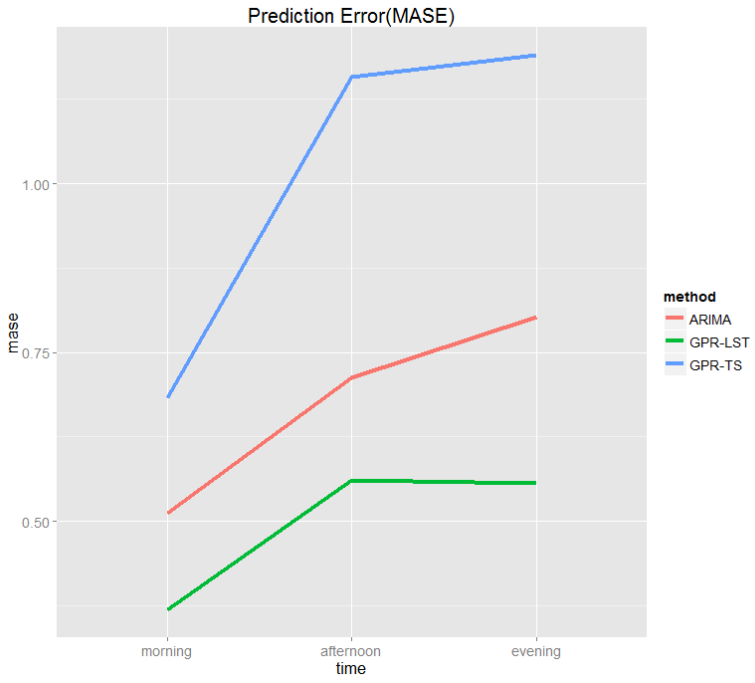
	NYC			Beijing		
	RMSE	MASE	MER	RMSE	MASE	MER
GPR-LST	33.175	0.481	0.096	13.432	0.611	0.125

Seasonal-ARIMA	45.384	0.678	0.133	15.925	0.707	0.146
GPR-Naive	71.865	0.909	0.185	18.779	0.843	0.170

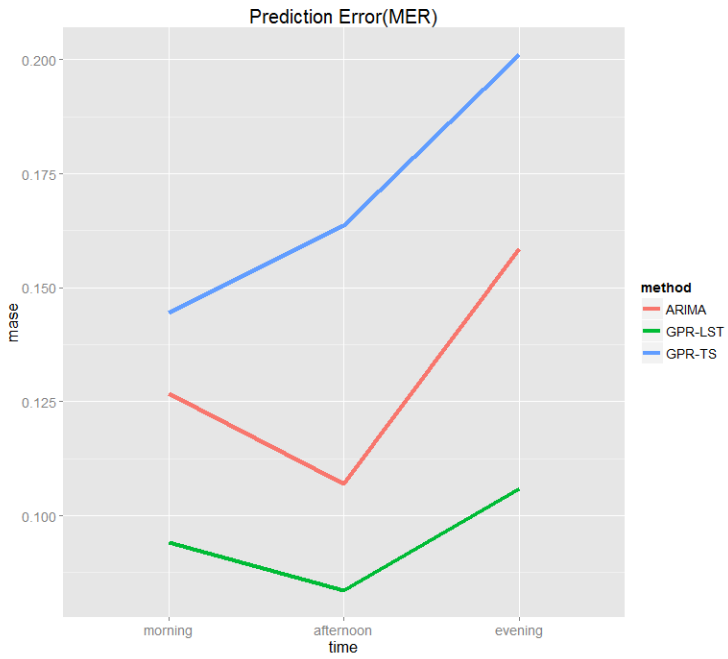
We further investigated the prediction errors of different methodologies at different time periods. We used NYC's outflow in the workdays as the main source for analysis. We divided a day into three main time periods, morning (6:00 am–11:59 am), afternoon (12:00 pm–17:59 pm), and evening (18:00 pm–23:59 pm) and plotted the prediction errors (MASE and MER) of different methodologies in Figure 6.3. From these plots, we can see that our proposed methodology (GPR-LST) performs best at any time period.

Apart from the advantage of our methodology, there are also some other interesting phenomena worth mentioning. The first one is that for both metrics, majority of the methodologies are more accurate in the morning compared with evening. The reason for this could be that people's mobility pattern in the morning is simpler and easier to be predicted since most people probably would just head to work places then. However, people's mobility pattern gets more complicated in the evening since they might go to restaurants, home, theaters, night clubs, etc., which makes an exact prediction more difficult. But for the prediction in the afternoon, two metrics show different trends. All methodologies had larger MASE but made smaller MER. We found that it is because

the flow volume across neighborhoods in the afternoon is usually stable while there are demand peaks in the morning and evening respectively (lots of people need to go to/get off work). Hence the naive one step prediction (the baseline of MASE) does a better job in the afternoon which results in the increase of the MASE value of all the prediction methodologies.



(a) MASE



(b) MER

Figure 6.3. Prediction error at different time periods

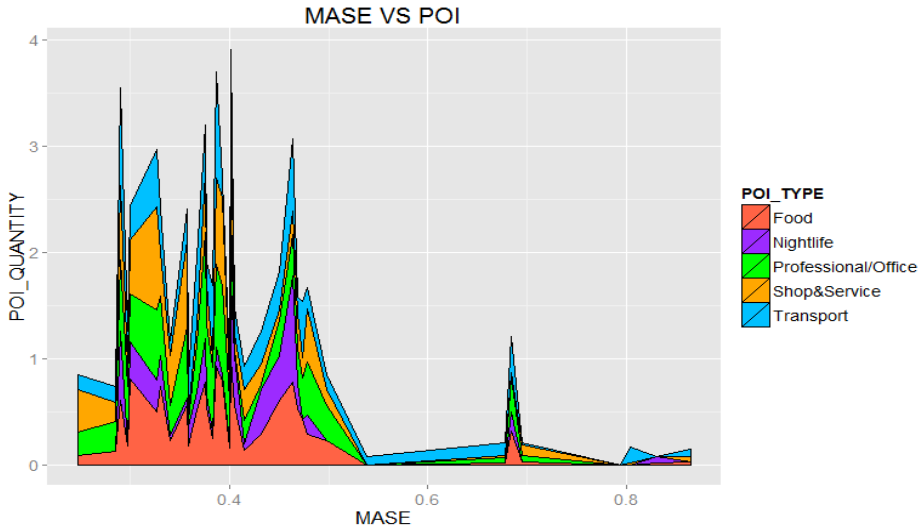
From the experiments above, we can see that our proposed methodology performs best, compared with some of the existing methodologies, and reduces the prediction error significantly. Furthermore, we assessed how our prediction methodology performed across different regions. More specifically, for each partitioned grid, we explored the relationship between the prediction error (MASE) of our methodology and the POI (point of interest) distribution. We collected NYC's POI data from the OpenStreetMap (OpenStreetMap 2017) and focused on 5 types of POIs: food, nightlife, professional/office, shop & service, transport. We do not consider the residential data here because the residential data in OpenStreetMap is very sparse and incomplete. Note

that the size of different POI types varies, e.g., in an office area, there could be more restaurants than actual offices. Hence, it is difficult to judge the function of a region based on the absolute number of POIs. To address this, we normalize the scale of each POI type in each partitioned grid into the range of (0,1) with:

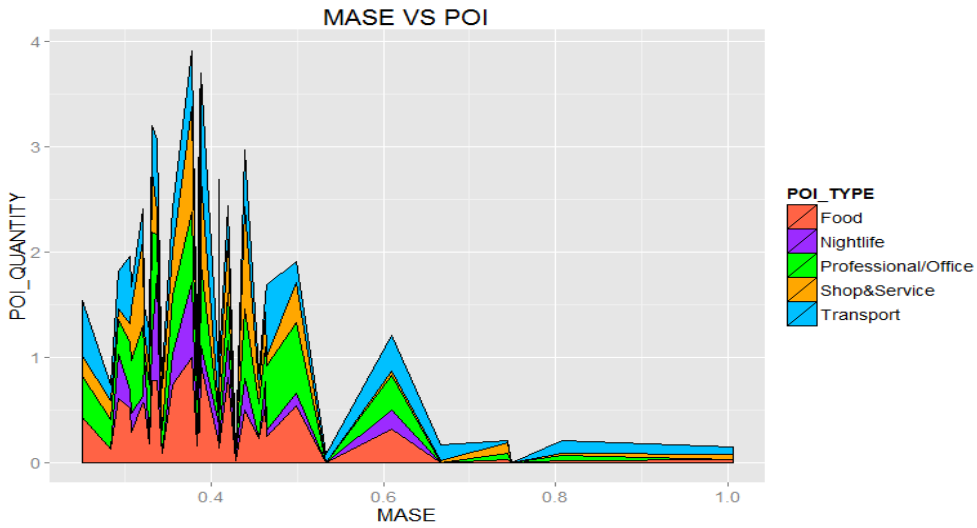
$$P_{i,k}^{'} = \frac{P_{i,k} - min_i(P_{i,k})}{max_i(P_{i,k}) - min_i(P_{i,k})} \qquad (6.6)$$

where $P_{i,k}$ is the number of POI of type k within grid i and $P_{i,k}^{'}$ is the normalized $P_{i,k}$. We plot the prediction error (MASE) and the normalized POI values of each grid in Figure 6.4. It is a stacked area plot where the x-axis indicates the MASE of our prediction methodology for different grids and the y-axis indicates the normalized value of different POIs in the corresponding grid. From the plot, we can see when there are certain amounts of POIs (the sum of normalized POI values is larger than a threshold, like 0.8) in an area, our prediction methodology generally makes less errors (the MASE is less than 0.5). This makes sense since in the urban areas with more POIs and more people's activities, the pattern of taxis' pick-ups and drop-offs tend to be more regular compared to suburban areas where people would take taxi less frequently and more randomly. But this relationship does not change smoothly. In other words, there is no strict increase/decrease function and some exceptions do exist. One reason for this is the inherent complication of human's mobility pattern, and many people usually do not take taxi frequently and regularly. Another reason could be that our collected POI data is not very complete, e.g., lack of residential data and the scale/popular of each POI is also not considered here, e.g.,

a big office POI like New York City Hall would definitely have a larger impact on the taxi demand than a POI of small company. Lastly, our sample is relatively small, with less than hundred grids in a city.



(a) Outflow



(b) Inflow

Figure 6.4 The prediction error (MASE) at different spatial units

Besides the number of POIs, we also explored the relationship between the number of passengers and prediction MASE in each area. The result is plotted in Figure 6.5, from which we can see there is a reciprocal relationship between them. When there are more people who took taxis in an area (more than 2500 pick-ups/drop-offs a day), our prediction methodology achieved quite high prediction accuracies (with MASE less than 0.5), confirming one of our hypotheses that when there are more human activities, it is easier to predict the number of pick-ups and drop-offs. But this relationship is also not a strict increase/decrease function.

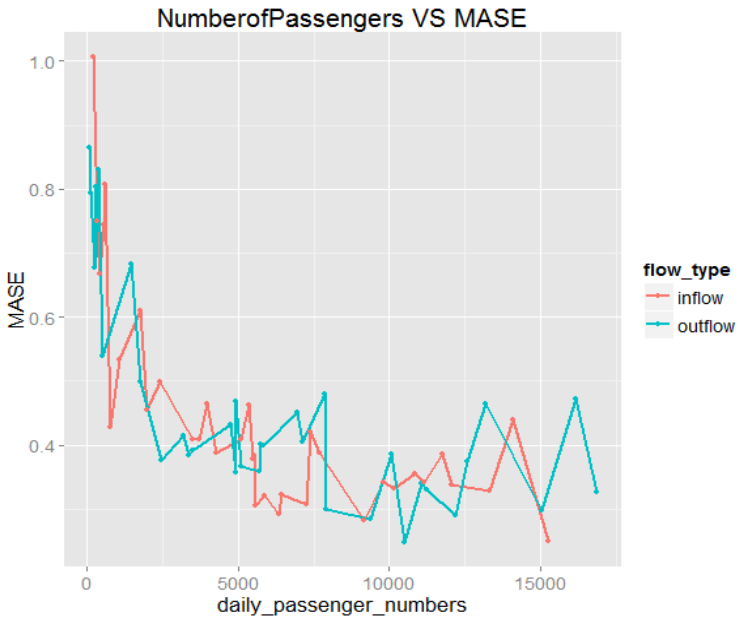
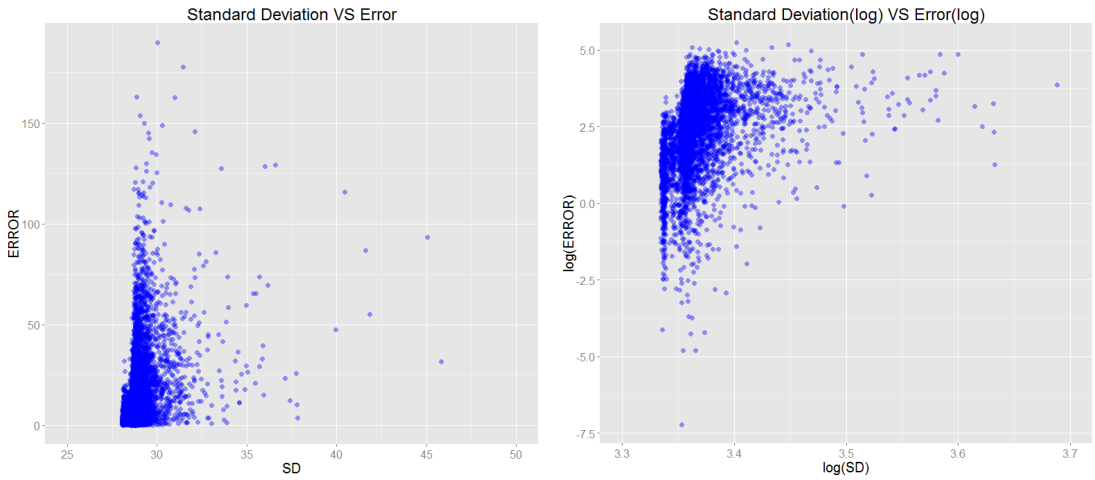


Figure 6.5 The number of pick-ups and drop-offs vs. prediction error (MASE)

Lastly we would also like to explore that for our proposed GPR-LST methodology, whether there is a relationship between the absolute prediction error and the standard deviation of the Gaussian Process Regression. We plot the distribution of absolute prediction error and the standard deviation in the Figure 6.6. From the plotting, it seems although in some cases the prediction error did increase as the standard deviation got larger, there is no strong relationship between them.



(a) Original Distribution (b) Distribution with Log Scale

Figure 6.6 Absolute Prediction Error vs Standard Deviation

6.3 The Flow Volume Between Neighborhoods

After the prediction of outflow (inflow) across the partitioned grids, we further clustered those grids with similar mobility pattern into neighborhoods and predict the flow volume between them. In particular, we clustered the grids with similar latent spatial features. Since each grid can be either an origin or a destination, we defined the mobility feature vector of grid i as:

$$\mathcal{S}_i = (\mathcal{S}_{o_i}, \mathcal{S}_{d_i}) \qquad (6.7)$$

and the distance between the two grids i and j as:

$$s_{ij} = |\mathcal{S}_i - \mathcal{S}_j|^{\alpha} * \left(\frac{\mathcal{S}_i * \mathcal{S}_j}{|\mathcal{S}_i| * |\mathcal{S}_j|}\right)^{\beta} \quad (6.8)$$

The left part is the Euclidean distance while the right part is the cosine between two spatial vectors. This distance function takes both direction and magnitude of the latent spatial features into account.

To cluster the grids with similar spatial latent features in neighborhoods, we adapted a bottom-up spatial hierarchical clustering approach. Specifically, in the beginning we assumed every grid is a neighborhood. Then we iteratively searched the pair of adjacent neighborhoods that have the smallest complete-linkage and merged them together. We repeated this merging procedure until certain criteria are met; for example, the smallest complete-linkage is larger than a given threshold. The clustered results of NYC and Beijing are shown in Figure 6.7 and Figure 6.8.

With the clustered neighborhoods, we can explore mobility patterns between them. For our analysis, we chose four representative neighborhoods: 1, 2, 6, and 12. We plotted their average volume of inflow and outflow in a day (see Figure 6.9). One notable common pattern among all four neighborhoods (but unrelated to neighborhood characteristics) is the drop of outflow volume between 3:00 pm and 4:00 pm that is caused by the shift switch of taxi drivers. We also observed that these four neighborhoods have very unique mobility patterns. The neighborhood 1 has the highest inflow peak in the morning at around 9:00 am, and the peaks of both inflow and outflow at around 7pm

– 8 pm, which indicates neighborhood 1 is an office district mixed with some residential functions; in fact, neighborhood 1 is mainly composed of financial district, one of the busiest business and tourist areas in New York City and many luxury apartments. On the other side, neighborhood 6, which is mainly composed of Upper West Side (an affluent, primarily residential area), has the highest peaks of outflow and inflow are in the morning and evening, respectively, which is a typical sign of residential district mixed with some other functions. Different from other areas, neighborhood 2 has significantly high volume of inflow in the evening, a sign of nightlife district. From these examples we can see that our extracted latent features generally distinguish different neighborhoods with diverse unique characteristics.

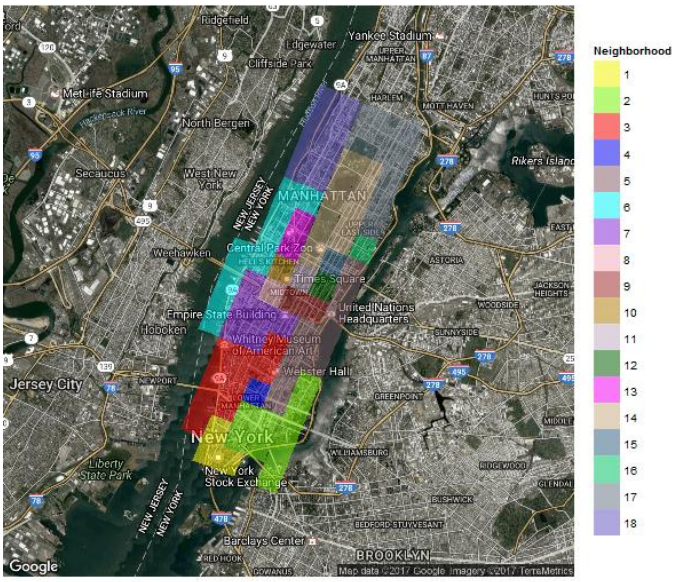


Figure 6.7 The clustered neighborhoods of NYC

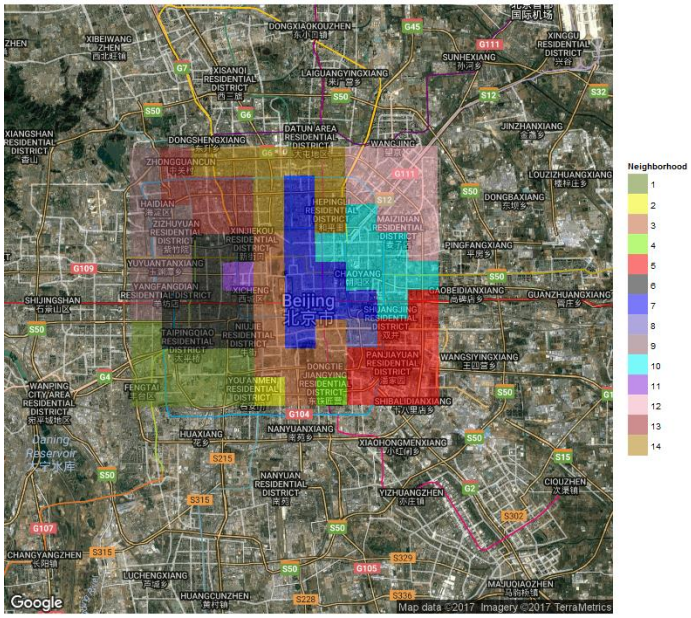
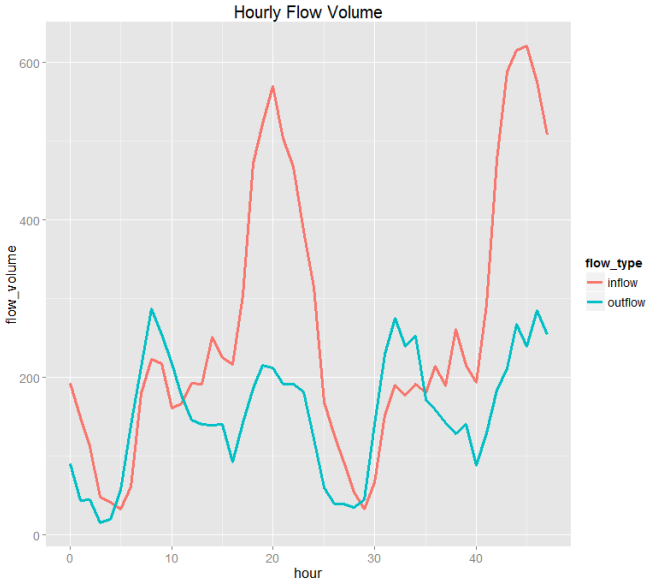


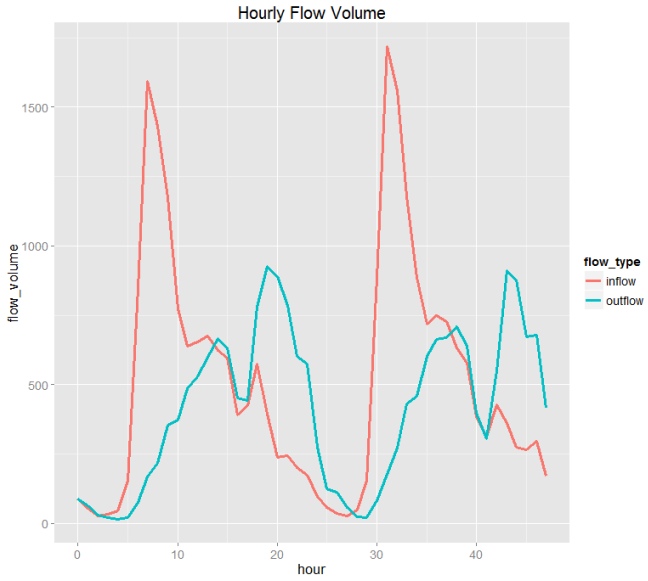
Figure 6.8 The clustered neighborhoods of Beijing

(a) Neighborhood-1



(b) Neighborhood-2

(c) Neighborhood-6



(d) Neighborhood-12

Figure 6.9 Average hourly inflow/outflow of selected neighborhoods

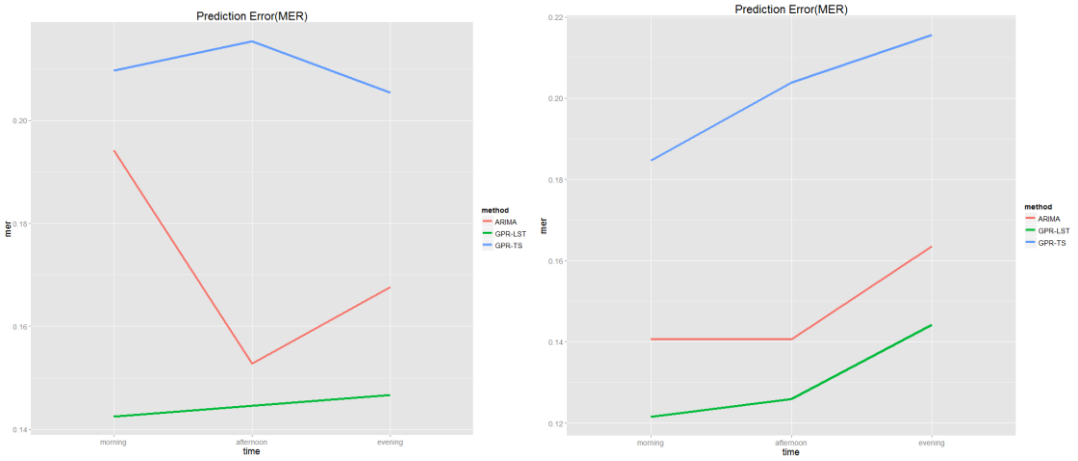
Based on the clustered neighborhoods, we would predict the flow volume between them using the method described in section 3.2.3. We also compared our methodology with the Seasonal-ARIMA and GPR-Naïve. For each pair of origin and destination neighborhoods, we trained a Seasonal-ARIMA model for it. As for GPR-Naïve, we trained one model with all the flow volume between any pair of neighborhoods.

We first compared the results between NYC and Beijing. From the table-4 we can see the proposed methodology achieves better prediction accuracy and reduces the prediction error by 15%-20% compared with others such as Seasonal-ARIMA.

Table 4: The prediction of flow volume between neighborhoods (NYC vs Beijing)

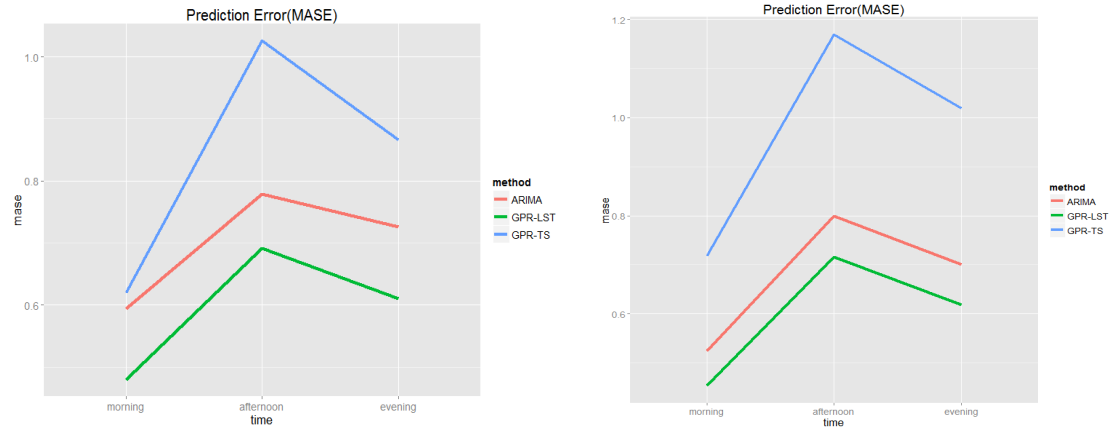
	NYC			Beijing		
	RMSE	MASE	MER	RMSE	MASE	MER
GPR-LST	6.766	0.586	0.144	8.9773	0.5848	0.1299
Seasonal-ARIMA	7.959	0.680	0.170	9.7870	0.6631	0.1473
GPR-Naive	9.843	0.815	0.209	22.0454	0.9486	0.2009

We also investigated how different methodologies perform in different time periods. Same as the previous section, we divided a day into three different time periods, morning, afternoon and evening. And we plotted the results in Figure 6.10 and Figure 6.11, which shows similar patterns as the previous section (the prediction of outflow/inflow), for example, most methodologies achieve better accuracy (less prediction error) in the morning compared with the evening. Because the flow volume in the afternoon has relatively stable temporal pattern compared with the ones in the morning and evening, all methods have higher MASE in the afternoon but less MER.



(a) NYC (b) Beijing

Figure 6.10 Prediction error(MER) at different time periods

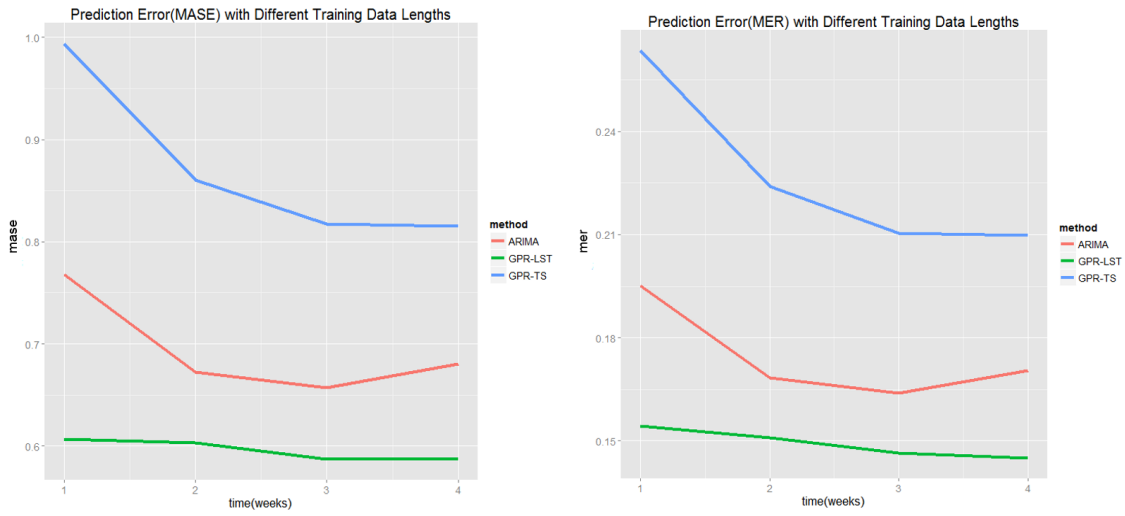


(a) NYC (b) Beijing

Figure 6.11: Prediction error (MASE) at different time periods.

We also investigated how different lengths of the training dataset would affect the prediction errors. Specifically, we trained each methodology with 1, 2, 3, 4 weeks data of NYC and used the next 2 weeks data for the verification. We plotted the results in the

Figure 6.12. From the figure we can see our proposed methodology achieves acceptable performance even with just 1 week's training data. And the prediction errors of all the methodologies become stable with 4 weeks' training data.



(a) MASE (b) MER

Figure 6.12 Prediction error with different Training Data Lengths

6.4 The Prediction of Popular Road Segments and Primary Origin/Destinations

Based on the predicted flow between neighborhoods, we further simulated the corresponding trajectory distributions in the road network and verified whether our synthetic trajectory distributions can accurately reflect the real traffic situation, and specifically, the hot road segments and their primary origins/destinations.

We mainly explored Beijing's taxi dataset in this section; for the New York City taxi dataset, there is no detailed trajectory of each trip, and we are not able to directly verify the correctness of our methodology. Since the taxi dataset of Beijing is a series of GPS points, for each trip we ran the Map-Matching algorithm proposed by (Newson and Krumm 2009) and projected the GPS points into a series of road segments that the taxi traveled through, in order to gain the ground truth.

We collected information on Beijing's road network from the OpenStreetMap. We converted the original OSM format into a nodes-edges graph with osm4routing (OSM4Routing 2017). We only kept the road segments within the boundary shown in Figure 6.8. and further removed those road segments that were only for pedestrians or bicycles. Eventually, 26,975 road segments and 20,334 intersections were left.

We first showed the accuracy of the top-K hot road segments prediction. Specifically, we predicted the top-5%, 10%, 15%,… of hot road segments based on the synthetic trajectory distributions in the next hour iteratively. We define the accuracy as:

$$accuracy\left(\hat{E}^k, E^k\right) = \frac{|\hat{E}^k \cap E^k|}{|E^k|} \quad (6.9)$$

where $\hat{E}^k$ is the predicted top k popular road segments and E^k is the actual top K popular road segments. We plotted the results of six models (shortest-path, top 3, top 6 shortest paths; top 1, top 3, and top 6 most likely paths) in Figure 6.13. From the figure, we can see that the shortest-path–based model achieves the lowest accuracy in most cases, and that the top-K likely based models inferred from the multivariate KDE perform

slightly better than the top-K shortest-path–based models—yet the advantage is not that significant. This could be caused by the sparsity of the data. In our collected dataset, there are usually just a few thousands trips each hour, which makes the statistical pattern of the trajectory distributions less regular. We might need to collect some more complete datasets in the future for further analysis. As we increase the value of K of the hot road segments, the accuracy of all models also increases and the accuracy difference between them gradually decreases. This is understandable since it becomes easier for all the models to predict the top-K hot road segments as we increase the value of K.

After the prediction of hot road segments, we attempted to further identify their formation through the origin or destination of the traffic in those road segments. Specifically, we tried to predict the top, top two, and top-K popular origin/destination neighborhoods of every road segment, based on the synthetic trajectory distributions. In other words, we wanted to see which neighborhood contributes largest (the second largest, third largest, and so on) amount of incoming/outgoing traffic volume for each road segment in the next hour. To measure the accuracy of the top-K primary origin/destination neighborhoods, we use a similar measurement metric as the previous top-K hot road segments:

$$accuracy\left(\hat{R}^k, R^k\right) = \frac{|\hat{R}^k \cap R^k|}{|R^k|} \quad (6.10)$$

where $\hat{R}^k$ is the predicted top k primary origin/destination neighborhoods while R^k is the actual top K primary origin/destination neighborhoods. Note that in the

experiment, we obtained the prediction accuracy for origin and destination neighborhoods separately, then used the mean as the corresponding accuracy. For example, the prediction accuracies of the top primary origin and destination neighborhoods are 0.72 and 0.71, respectively. As a result, the prediction accuracy of the top origin/destination neighborhood is (0.72 + 0.71) / 2 = 0.715. The final result is plotted in Figure 6.14. From Figure 6.14 we can see that the top-K likely-path–based models also achieve better prediction accuracies, as compared with the top-K shortest paths based models, and that the advantage is more obvious. In contrast to the prediction of hot road segments, the top likely-path–based model performs best, while the top-6 shortest-path–based model performs the worst in most cases. As K increases, all of the models generally achieve higher accuracy for the prediction of the K primary origin/destination neighborhoods; yet in the beginning, the prediction accuracy decreases. We found that one reason for this finding is because a road segment is usually visited more frequently by the vehicles starting from or ending at that corresponding neighborhood. As a result, the prediction of the top primary origin/destination neighborhood is relatively easier. It becomes difficult to predict the second, third, … primary neighborhoods, as there are more possibilities from which to choose.

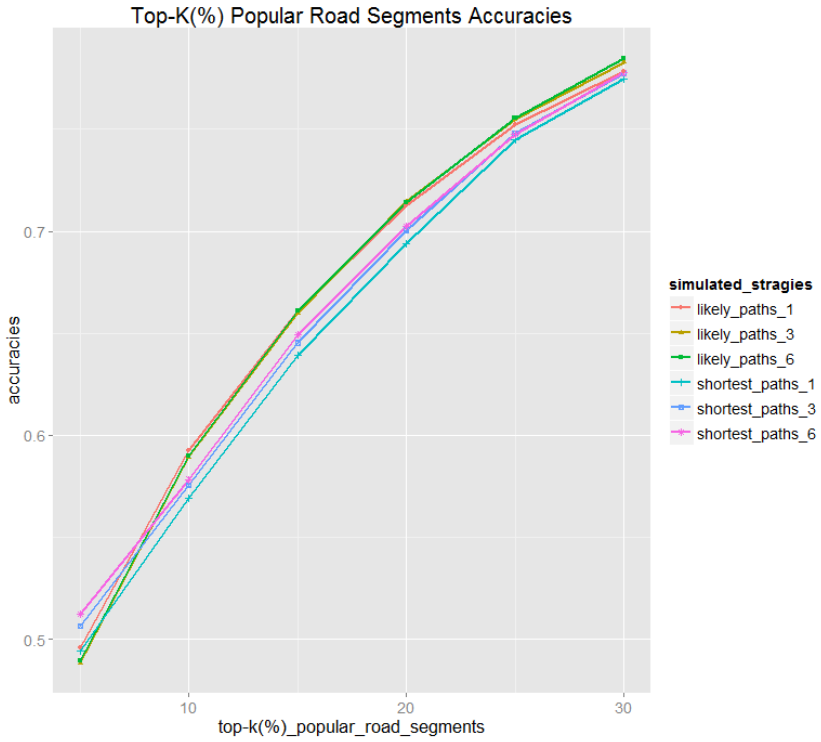


Figure 6.13 Prediction of hot road segments.

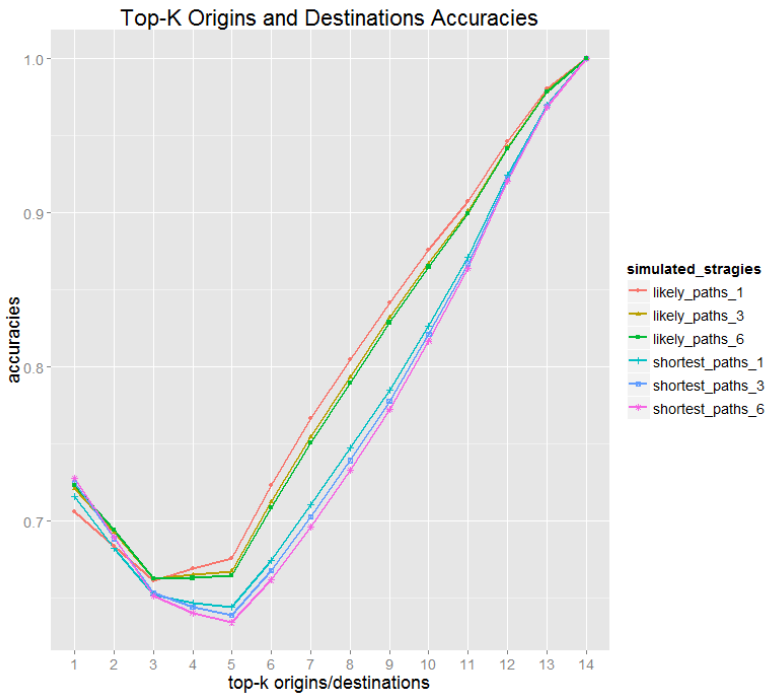


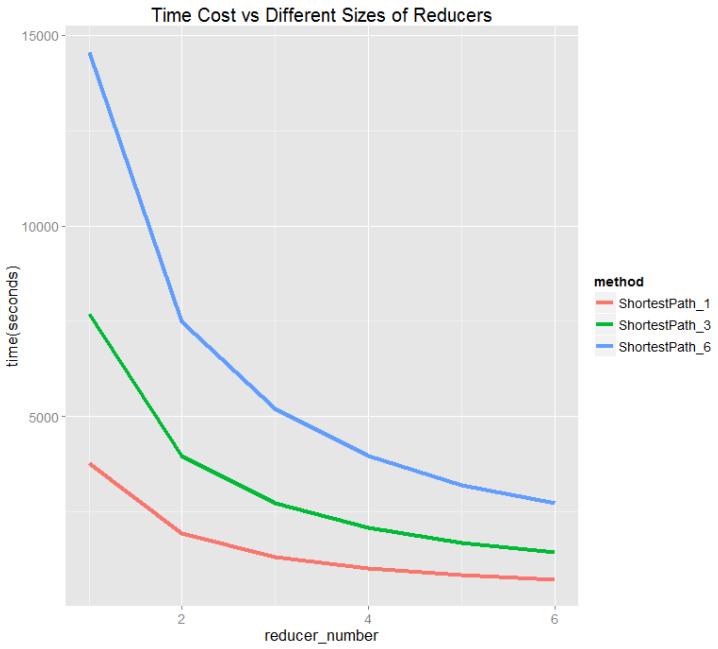
Figure 6.14 Prediction of Top-K origin/destination neighborhoods.

6.5 Time Performance of Distributed Trajectory Distribution Simulation Algorithms

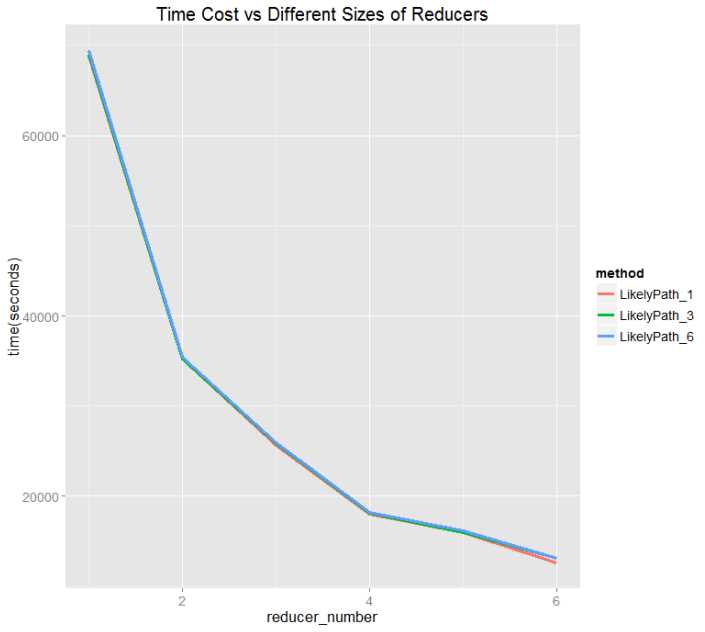
Finally, we demonstrated the scalability of our designed MapReduce-based trajectory distribution simulation algorithms. We conducted our experiments on a Hadoop cluster composed of six machines. Each machine in the cluster had an Intel Xeon 2.2GHz 4 Core CPU with 48 GB RAM and a 1 TB hard drive at 7200 rpm. There is one named node and six data nodes in our cluster (the named node is also a data node). The version of Hadoop is 2.7.1.

We can see from Algorithm 5.1 that the Map phase is pretty straightforward. We simply sent a few hundred records of flow volumes between neighborhoods to mappers

and they generate the corresponding flow volume between each pair of edges, which costs just 1–3 minutes in our cluster. On the other hand, the Reduce phase is computationally intensive, as it is the core of the trajectory distribution simulation. As a result, we mainly show the running time of our program versus the increasing number of reducers in Figure 6.15. From Figure 6.15, we can see that the running time of the program decreases gradually as the number of reducers increases, which demonstrates the scalability of our designed algorithms. Note that since the reduce phase is computationally intensive and our Hadoop cluster is relatively small (with only six machines), it can only run up to six reducers at one time. As a result, adding additional reducers will not help improve time performance. For the top-K shortest-path–based models, the time cost of the program also increases as the value of K gets larger, which is reasonable since there are more potential routes to be searched. As for the top-K likely-path–based models, there is no significant difference for different K values, because we generally need to search all the potential routes until we reach a certain threshold (as shown in line 14 of Algorithm 5.2).



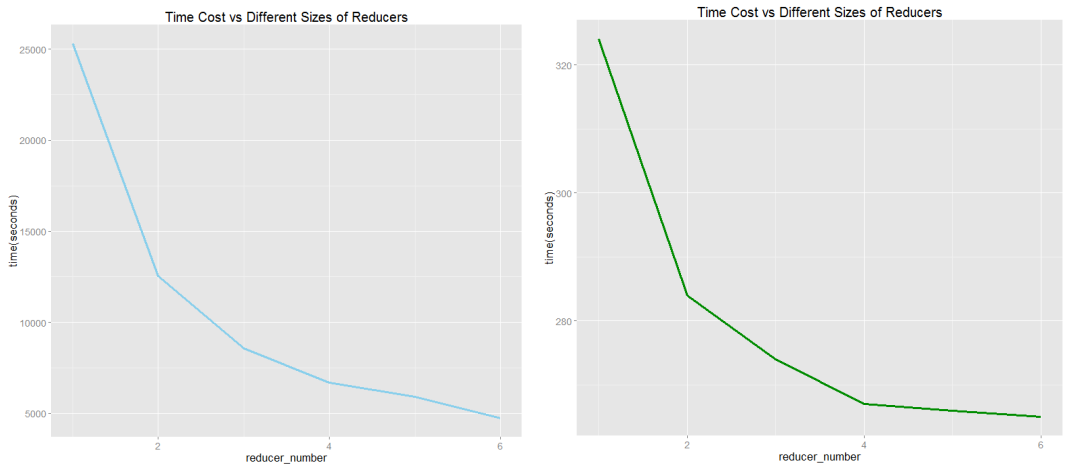
(a) Top-K shortest paths



(b) Top-K likely paths

Figure 6.15 Running time of trajectory distribution simulation vs number of reducers.

We also explored the time performance of the prediction of the top-k hot road segments and the primary origin/destination neighborhoods. For the prediction of the primary origin/destination neighborhoods, we randomly chose a road segment and ran the program based on the synthetic trajectory distribution. The results are shown in Figure 6.16, and they both also showed good scalability.



(a) Popular road segments (b) Primary OD neighborhoods

Figure 6.16 Running time of trajectory distribution analysis versus the number of reducers.

7.0 LIMITATIONS

Our research has provided new methods and insights into learning mobility patterns that can be applied to different applications. However, there are limitations to the research described in this thesis, discussed briefly below.

Our model extracts the latent spatial and temporal features from datasets to predict mobility patterns. Our current model is limited to normal mobility activities and does not take into account deviation from these activities. For example, our model cannot predict mobility based on abnormal events, which could dramatically change people's daily mobility pattern, such as a NFL football game, a national holiday, or extreme weather are not handled by our model.

Our methods for the trajectory distribution simulation only consider distance for route finding. While distance is a predominant criterion for finding routes, there are other criteria, such as travel time and least tolls, that are important as well.

The experiments, to validate our proposed methodology, were focused on taxi data only. For this, our prediction results and conclusions are only valid for mobility patterns through taxi activities and not other mobility activities..

8.0 CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we propose to predict human spatial-temporal mobility at a large scale. Specifically, this thesis has several major components. Firstly we designed a latent feature based methodology for the prediction of spatial-temporal activities such as the outflow/inflow of the vehicles of each neighborhood. Specifically, we modeled people's spatial-temporal fluxes as a tensor and extract the latent spatial-temporal features through factorization. Then, we mathematically modeled the relationship between those extracted latent features and human mobility with a Gaussian process regression for future prediction. Compared with the existing techniques such as ARIMA, the designed methodology can inherently consider the characteristics of both spatial and temporal features of the predicted activities.

After that, we further predicted the vehicle trajectory distributions in the road network at a city level, from which the hot road segments and their formation can be predicted and identified in advance, such as which road segments will have high traffic volume, along with the origins and destinations of the majority of the traffic in those hot road segments. The vehicle trajectory distribution prediction comprised three steps: (1) a methodology for the prediction of flow between neighborhoods that combined both latent and explicit features; (2) different models for the simulation of the corresponding flow trajectory distributions in the road network, from which the hot road segments and their

formation can be predicted and identified in advance; and (3) different efficient MapReduce-based distributed algorithms for the real-time simulation and analysis for large-scale simulation of trajectory distributions.

To verify the proposed methodology in this thesis, we conducted two case studies on Beijing and New York City's taxi trip data with a series of experiments. For the prediction of people's outflow, inflow, and the flow between neighborhoods, the results showed that our designed methodology achieves a high degree of accuracy. Prediction errors are reduced significantly, as compared with some existing methodologies, such as Seasonal-ARIMA. Given the predicted flow between neighborhoods, we further simulated their trajectory distributions in the road network. Based on that, we predicted the top-K hot road segments and the primary origin/destination neighborhoods of the traffic passing through the hot road segments of interest. The results showed that our synthetic trajectory distributions accurately reflected the overall traffic situation. For example, for the prediction of the top 15% hot road segments, our methodology generally achieves an accuracy of around 65%. However, different models have different performances under different situations. For example, for the prediction of primary origin/destination neighborhoods, the top-K likely-path–based models inferred from multivariate KDE achieves a higher degree of accuracy, compared with the top-K shortest-path–based models; but for the prediction of hot road segments, their advantage is not that significant. More experiments may be done in the future to explore how

different models perform under different conditions, so that people could choose the right model based on their specific needs.

Finally, we explored the time performance of our designed MapReduce based algorithms on a Hadoop cluster consisting of six servers. The results show that as the number of reducers goes up, the time cost of our program goes down gradually, which demonstrated the scalability of our algorithm.

With regard to future research directions, there are several topics we can explore. First, in this thesis we predict the dynamic betweenness centrality of each road segment, and identify the hot road segments based on it. In the future we could further predict the average speed of each road segment based on the dynamic betweeness centrality, given the average speed is a more intuitive indicator of potential traffic congestion. Second, here we propose two models for the trajectory distribution simulation including the top-K shortest paths based model and top-K likely paths based model. Although both of them show good accuracy, we can try to design some more accurate models which take more factors into consideration, for example, the features of each road segment (the number of lanes, whether it is a highway or not, etc.), and estimate the possibility of each route. Another future work we can do is to detect the abnormal events and analyze the potential causes based on the synthetic trajectory distribution. Specifically, we can detect the road segments which would have significantly higher (or lower) traffic volume compared with the historical values, and identify the corresponding causes such as which neighborhood

contributes significantly more (or less) incoming/ongoing traffic. We can further extract the feeds from some location based social network and describe what happens.

BIBLIOGRAPHY

Akdogan, A., U. Demiryurek, F. Banaei-Kashani and C. Shahabi (2010). Voronoi-based geospatial query processing with mapreduce. Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, IEEE.

Castro, P. S., D. Zhang and S. Li (2012). Urban traffic modelling and prediction using large scale taxi GPS traces. International Conference on Pervasive Computing, Springer.

Chen, C., J. Hu, Q. Meng and Y. Zhang (2011). Short-time traffic flow prediction with ARIMA-GARCH model. Intelligent Vehicles Symposium (IV), 2011 IEEE, IEEE.

Chen, L., M. Lv and G. Chen (2010). "A system for destination and future route prediction based on trajectory mining." Pervasive and Mobile Computing **6**(6): 657-676.

Chen, P.-T., F. Chen and Z. Qian (2014). Road traffic congestion monitoring in social media with hinge-loss Markov random fields. 2014 IEEE International Conference on Data Mining, IEEE.

Chen, Z., H. T. Shen and X. Zhou (2011). Discovering popular routes from trajectories. 2011 IEEE 27th International Conference on Data Engineering, IEEE.

Clark, S. (2003). "Traffic prediction using multivariate nonparametric regression." Journal of transportation engineering **129**(2): 161-168.

Comito, C., D. Falcone and D. Talia (2015). Mining Popular Travel Routes from Social Network Geo-Tagged Data. Intelligent interactive multimedia systems and services, Springer**:** 81-95.

Cranshaw, J., R. Schwartz, J. I. Hong and N. Sadeh (2012). The livehoods project: Utilizing social media to understand the dynamics of a city. International AAAI Conference on Weblogs and Social Media.

Davis, G. A. and N. L. Nihan (1991). "Nonparametric Regression and Short‐Term Freeway Traffic Forecasting." Journal of Transportation Engineering.

De Lathauwer, L., B. De Moor and J. Vandewalle (2000). "On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors." SIAM Journal on Matrix Analysis and Applications **21**(4): 1324-1342.

Dean, J. and S. Ghemawat (2008). "MapReduce: simplified data processing on large clusters." Communications of the ACM **51**(1): 107-113.

Deri, J. A., F. Franchetti and J. M. Moura (2016). Big Data computation of taxi movement in New York City. Proceedings of the 1st IEEE Big Data Conference Workshop on Big Spatial Data.

Deri, J. A. and J. M. Moura (2015). Taxi data in New York City: a network perspective. Signals, Systems and Computers, 2015 49th Asilomar Conference on, IEEE.

Eldawy, A., Y. Li, M. F. Mokbel and R. Janardan (2013). CG_Hadoop: computational geometry in MapReduce. Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM.

Eldawy, A. and M. F. Mokbel (2013). "A demonstration of SpatialHadoop: an efficient mapreduce framework for spatial data." Proceedings of the VLDB Endowment **6**(12): 1230-1233.

Ferreira, N., J. Poco, H. T. Vo, J. Freire and C. T. Silva (2013). "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips." Visualization and Computer Graphics, IEEE Transactions on **19**(12): 2149-2158.

Franses, P. H. (2016). "A note on the Mean Absolute Scaled Error." International Journal of Forecasting **32**(1): 20-22.

Froehlich, J. and J. Krumm (2008). Route prediction from trip observations, SAE Technical Paper.

Froehlich, J., J. Neumann and N. Oliver (2009). Sensing and Predicting the Pulse of the City through Shared Bicycling. IJCAI.

Gao, S., Y. Liu, Y. Wang and X. Ma (2013). "Discovering spatial interaction communities from mobile phone data." Transactions in GIS **17**(3): 463-481.

Guo, D., S. Liu and H. Jin (2010). "A graph-based approach to vehicle trajectory analysis." Journal of Location Based Services **4**(3-4): 183-199.

Guo, Q., B. Palanisamy and H. A. Karimi (2014). A distributed polygon retrieval algorithm using MapReduce. Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on, IEEE.

Han, B., L. Liu and E. Omiecinski (2015). "Road-network aware trajectory clustering: Integrating locality, flow, and density." IEEE Transactions on Mobile Computing **14**(2): 416-429.

Hong, L., Y. Zheng, D. Yung, J. Shang and L. Zou (2015). Detecting urban black holes based on human mobility data. Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM.

Hua, C.-i. and F. Porell (1979). "A critical review of the development of the gravity model." International Regional Science Review **4**(2): 97-126.

Jeung, H., M. L. Yiu, X. Zhou and C. S. Jensen (2010). "Path prediction and predictive range querying in road network databases." The VLDB Journal **19**(4): 585-602.

Ji, C., T. Dong, Y. Li, Y. Shen, K. Li, W. Qiu, W. Qu and M. Guo (2012). Inverted grid-based knn query processing with mapreduce. ChinaGrid Annual Conference (ChinaGrid), 2012 Seventh, IEEE.

Jiang, S., J. Ferreira Jr and M. C. Gonzalez (2012). Discovering urban spatial-temporal structure from human activity patterns. Proceedings of the ACM SIGKDD international workshop on urban computing, ACM.

Kaltenbrunner, A., R. Meza, J. Grivolla, J. Codina and R. Banchs (2010). "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system." Pervasive and Mobile Computing **6**(4): 455-466.

Kamath, K. Y., J. Caverlee, Z. Cheng and D. Z. Sui (2012). Spatial influence vs. community influence: modeling the global spread of social media. Proceedings of the 21st ACM international conference on Information and knowledge management, ACM.

Kolda, T. G. and B. W. Bader (2009). "Tensor decompositions and applications." SIAM review **51**(3): 455-500.

Lam, H. T. and E. Bouillet (2014). Online event clustering in temporal dimension. Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM.

Lathia, N., D. Quercia and J. Crowcroft (2012). The hidden image of the city: sensing community well-being from urban mobility. International Conference on Pervasive Computing, Springer.

Li, X., J. Han, J.-G. Lee and H. Gonzalez (2007). Traffic density-based discovery of hot routes in road networks. International Symposium on Spatial and Temporal Databases, Springer.

Liu, M., K. Fu, C.-T. Lu, G. Chen and H. Wang (2014). A search and summary application for traffic events detection based on twitter data. Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM.

Liu, S., Y. Liu, L. M. Ni, J. Fan and M. Li (2010). Towards mobility-based clustering. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.

Liu, X. and H. A. Karimi (2006). "Location awareness through trajectory prediction." Computers, Environment and Urban Systems **30**(6): 741-756.

Liu, X., Y. Zhu, Y. Wang, G. Forman, L. M. Ni, Y. Fang and M. Li (2012). "Road recognition using coarse-grained vehicular traces." HP Labs, HP Labs2012.

Liu, Y., X. Liu, S. Gao, L. Gong, C. Kang, Y. Zhi, G. Chi and L. Shi (2015). "Social sensing: A new approach to understanding our socioeconomic environments." Annals of the Association of American Geographers **105**(3): 512-530.

Liu, Y., F. Wang, Y. Xiao and S. Gao (2012). "Urban land uses and traffic 'source-sink areas': Evidence from GPS-enabled taxi data in Shanghai." Landscape and Urban Planning **106**(1): 73-87.

Matthias, H.-P. K. M. R. and S. A. Zuefle (2008). "Statistical density prediction in traffic networks."

Neill, D. B. (2009). "Expectation-based scan statistics for monitoring spatial time series data." International Journal of Forecasting **25**(3): 498-517.

Newson, P. and J. Krumm (2009). Hidden Markov map matching through noise and sparseness. Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, ACM.

Nishi, K., K. Tsubouchi and M. Shimosaka (2014). Hourly pedestrian population trends estimation using location data from smartphones dealing with temporal and spatial sparsity. Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM.

Noulas, A. and C. Mascolo (2013). Exploiting foursquare and cellular data to infer user activity in urban environments. Mobile Data Management (MDM), 2013 IEEE 14th International Conference on, IEEE.

Noulas, A., S. Scellato, C. Mascolo and M. Pontil (2011). "Exploiting Semantic Annotations for Clustering Geographic Areas and Users in Location-based Social Networks." The Social Mobile Web **11**.

NYCOpenData. (2016). "NYC Open Data." Retrieved 01/01, 2016, from https://opendata.cityofnewyork.us/.

OpenStreetMap. (2017). Retrieved 03/01, 2017, from https://www.openstreetmap.org/.

OSM4Routing. (2017). "OSM4Routing." from https://github.com/Tristramg/osm4routing.

Patricia S. Hu, T. R. (2001). 2001 National Household Travel Survey. New York Add-On, New York City – New York County/Manhattan.

Puri, S., D. Agarwal, X. He and S. K. Prasad (2013). MapReduce algorithms for GIS polygonal overlay processing. Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International, IEEE.

Quercia, D., L. M. Aiello, R. Schifanella and A. Davies (2015). The digital life of walkable streets. Proceedings of the 24th International Conference on World Wide Web, ACM.

Rasmussen, C. E. (2006). "Gaussian processes for machine learning."

Reades, J., F. Calabrese and C. Ratti (2009). "Eigenplaces: analysing cities using the space–time structure of the mobile phone network." Environment and Planning B: Planning and Design **36**(5): 824-836.

Ren, Y., M. Ercsey-Ravasz, P. Wang, M. C. González and Z. Toroczkai (2014). "Predicting commuter flows in spatial networks using a radiation model based on temporal ranges." arXiv preprint arXiv:1410.4849.

Sayyadi, H., M. Hurst and A. Maykov (2009). Event detection and tracking in social streams. Icwsm.

Scellato, S., M. Musolesi, C. Mascolo, V. Latora and A. T. Campbell (2011). NextPlace: a spatio-temporal prediction framework for pervasive systems. Pervasive computing, Springer**:** 152-169.

Shekhar, S. and B. Williams (2008). "Adaptive seasonal time series models for forecasting short-term traffic flow." Transportation Research Record: Journal of the Transportation Research Board(2024): 116-125.

Simonoff, J. (1996). Smoothing methods in Statistics. 1996. Cité en**:** 163.

Toole, J. L., M. Ulm, M. C. González and D. Bauer (2012). Inferring land use from mobile phone activity. Proceedings of the ACM SIGKDD international workshop on urban computing, ACM.

Wang, F., R. Lee, Q. Liu, A. Aji, X. Zhang and J. Saltz (2011). Hadoop-gis: A high performance query system for analytical medical imaging with mapreduce, Technical report, Emory University.

Wang, S., F. Li, L. Stenneth and S. Y. Philip (2016). Enhancing Traffic Congestion Estimation with Social Media by Coupled Hidden Markov Model. Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer.

Wang, Y., Y. Zheng and Y. Xue (2014). Travel time estimation of a path using sparse trajectories. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.

Wei, L.-Y., Y. Zheng and W.-C. Peng (2012). Constructing popular routes from uncertain trajectories. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.

Williams, B. M. and L. A. Hoel (2003). "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results." Journal of transportation engineering **129**(6): 664-672.

Wilson, A. G. (1967). "A statistical theory of spatial distribution models." Transportation research **1**(3): 253-269.

WorldTradeCenter. (2017). "ONE WORLD TRADE CENTER." from https://www.wtc.com/about/buildings/1-world-trade-center.

Yen, J. Y. (1970). "An algorithm for finding shortest routes from all source nodes to a given destination in general networks." Quarterly of Applied Mathematics: 526-530.

Yu, X., H. Zhao, L. Zhang, S. Wu, B. Krishnamachari and V. O. Li (2010). Cooperative sensing and compression in vehicular sensor networks for urban monitoring. Communications (ICC), 2010 IEEE International Conference on, IEEE.

Yuan, J., Y. Zheng and X. Xie (2012). Discovering regions of different functions in a city using human mobility and POIs. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.

Zhang, F., D. Wilkie, Y. Zheng and X. Xie (2013). Sensing the pulse of urban refueling behavior. Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, ACM.

Zhang, K., Y.-R. Lin and K. Pelechrinis (2016). EigenTransitions with Hypothesis Testing: The Anatomy of Urban Mobility. Tenth International AAAI Conference on Web and Social Media.

Zhang, W., L. Zhang, Y. Ding, T. Miyaki, D. Gordon and M. Beigl (2011). Mobile sensing in metropolitan area: Case study in beijing. Mobile Sensing Challenges Opportunities and Future Directions, Ubicomp2011 workshop.

Zheng, Y., T. Liu, Y. Wang, Y. Zhu, Y. Liu and E. Chang (2014). Diagnosing New York city's noises with ubiquitous data. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM.

Zheng, Y., Y. Liu, J. Yuan and X. Xie (2011). Urban computing with taxicabs. Proceedings of the 13th international conference on Ubiquitous computing, ACM.

Zhou, X., A. V. Khezerlou, A. Liu, Z. Shafiq and F. Zhang (2016). A traffic flow approach to early detection of gathering events. Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM.

Zhu, H., J. Luo, H. Yin, X. Zhou, J. Z. Huang and F. B. Zhan (2010). Mining trajectory corridors using Fréchet distance and meshing grids. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer.