龐　岩博

博士論文

Development of an Agent Model for People Flow Simulation

Based on Reinforcement Learning Approach

（強化学習に基く人の流れのシミュレーションのためのエージェントモデルの構築とその応用）

龐　岩博

"To my parents, partner and grandparents"

Yanbo Pang

THE UNIVERSITY OF TOKYO

Abstract

Department of Civil Engineering

Doctor of Philosophy

Development of an Agent Model for People Flow Simulation Based on Reinforcement Learning Approach
(強化学習に基づく人の流れのシミュレーションのためのエージェントモデルの構築とその応用)

by Yanbo PANG

Understanding individual and crowds dynamics in urban environments is critical for numerous applications, such as urban planning, traffic forecasting and location-based services. For example, monitoring dynamic population distribution and foreseeing crowds density are the fundamental for urban planners to design and improve public space for congestion reduction and evacuation guidance. Decision makers may need to forecast what will happen if new policies are introduced and how individual and group behaviors will change.

In the past few decades, travel demand modeling and simulation approach have been the most widely applied for capturing urban dynamics in the city-wide level. They support complex urban and transportation planning, as well as management tasks on different levels of granularity in space and time. For a long period of time, discrete choice models of human activities and decision making in travel-related choices were used as agent behavioral models. However, developing such models in the common scenario require details of demographic attributes and active trip purpose reports from travelers, and data source to provide such information is the National Household Survey, which is updated synchronously at infrequent intervals, and costs money and time. Thus, the applications of this method are limited in scenarios and the areas where surveys are conducted.

On the other hand, with the explosion of information and communications technology (ICT) and Internet-of-Things technologies, emerging data collection methods have enabled researchers to unravel individual mobility pattern and to generate models that could reproduce the time-varying characteristics in human trajectories. For example, high quality geolocated data such as call detailed records (CDRs), GPS, and social media data have quickly overtaken traditional high-cost data (i.e., census and travel surveys) as major data resources and have promoted a series of data-driven approaches to support transportation management, congestion management and disaster response. However, because of the strict privacy policy, researchers had to face a trade-off between developing fine-grained individual level modeling but hard to be generalized to large scale population, or leveraging large amount anonymous data to derive citywide population mobility on aggregated level. Although the crowd-sourced locational data are already used for agent-based mobility simulation to improve accuracy, no existing work provides bottom-up approach for modeling and simulating human mobility using emerging locational big data.

In this thesis, we develop a novel reinforcement learning based agent model which is capable of reproducing individual's daily travel behavior from anonymous locational data. To do so, we first model people's daily travel behavior as sequential decision makings under Markov Decision Process (MDP), which is the fundamental framework of reinforcement learning. We extend the application of reinforcement learning to real world by multiple steps. First, we designed the urban environment model which provides sufficient and accurate state-action space that efficiently decrease the calculation cost. Then we discuss the algorithms for finding optimal policy. To have a robust and fast training process, we explored several value-based and policy based algorithms including tabular solution methods and approximation methods with the power of neural network. Another challenge is how to derive human behavior preferences from passively collected location data. We introduce inverse reinforcement learning techniques that are capable of recovering reward functions from demonstration trajectories. To achieve this goal, we propose a data pre-processing pipeline to extract individual's daily trip with transport mode from raw GPS data. This method overcomes the data sparsity issue and can be applied to a variety of mobility datasets. By inferring and implementing the behavior preference parameters to agent model, the agents are capable of of learning mobility sequences from raw locational data while incorporating behavioral parameters that are sensitive to environmental context is introduced.

We applied the models to the data collected from GPS data collected by a smartphone application and People Flow Data (a data processed from Person Trip Survey which consists of spatio-temporal information) in urban area in Japan. The agent models are validated and compared with existing behavior models. The results show that proposed model framework outperforms the straightforward reinforcement learning, with the power of learning from

demonstration data. On the other hand, because the high flexibility of agent models, it is difficult to evaluate the generate trajectories. To better evaluate the proposed modeling framework, we simulate daily people movement based on the developed model and compared the simulation results with ground truth. Experiments are both launched on normal day and disaster scenarios.

Another important application of the proposed modeling and simulation framework is to forecast people movement in unprecedented scenario. Lack of historical data is the major challenge for direct forecasting because the collection and storage of emerging data sets are just started from recent years. We focus on the case that the same event (i.e. disasters) have happened at other places, where the data of people movement is sensed and stored. We use the mobility data collected from other places and develop the agent models, then simulate people movement in target area.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Sekimoto Yoshihide for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. I would not have the opportunity to start my study in this exciting and meaningful topic. Thank you so much for spending time revising all my papers and presentation. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Fuse Takashi, Prof. Oguchi Takashi, Prof. Shimosaka Masamichi, and Dr. Yoshimura Kei, for their insightful comments, technical guidance and hard questions.

I would like to show my thank to all the lab members, who supported me at every bit and without whom it was impossible to accomplish the end task. I will always remember all the help from Dr. Kashiyama, who helps me overcome all the technical problems and encourages me when I get depressed. Dr. Kanasugi, always prepares perfect and easy-to-use data that really makes my work easy and smooth. Dr. Seto, who gives me a lot of advice about how to write paper, present my study and support my life in the lab. I will also not forget Dr.Sudo, although we have just spent one year together, you guide me to the machine learning and make me realize what topic I am interested in. Dr.Fujiwara, every time you spend hours to listen and consider my research topics and help me to improve my paper, without you that paper will not be accepted.

I would also like to thank to my co-authors Dr. Tsubouchi and Dr. Yabe. Being a member of the joint research has been a wonderful experience. Thank you for all the wonderful ideas and excellent comments from the weekly meeting. In particular, I am grateful to you for enlightening me the first glance of research.

Also I thank my friends in the University of Tokyo: Maeda Hiroya, Suseno Wangsit Wijita, Shiho Hosaka, Sakata Riko, Maeda Midori, Sobue Hideaki, Yokozaka Naoto, Fan Zipei, Jiang Renhe. I will always remember the times we travel, play and discuss. You made my life in Japan happy and meaningful.

My sincere thanks also goes to our lab secretary, Danjo Ryoko and Homma Rieko, for supporting me all the paperwork and procedures. Ms. Aoyama at Civil Engineering office, who were patient when I made mistakes in submitting materials.

Last but not the least, I would like to thank my family: my parents for giving supporting me spiritually throughout my life; my partner, you light up my life. I hope I would make you proud of me.

Contents

List of Figures

List of Tables

For/Dedicated to/To my…

1 Introduction

1.1 Background

In the last few decades, the rapid urbanization and increase of population have challenge urban planers in various of issues such as traffic congestion, longer commuting, accidents, loss of public space and disaster management. Although the major challenge is still the gap between increasing demand and current supply for most regions and countries, it can be no longer solved easily by the solutions such as expanding roadway, adding stations or constructing new railway lines because the the investment and space is limited in current age. How to accurately capture the demand and leverage existing infrastructures effectively is essential for urban planning and management.

On the other hand, the sake of travelers' expectations of seamless travel and for that of mobility service providers' pursuit of efficient solutions, the attention of studying travel demand have shifted from the citywide population level to individual level. The business successes of emerging mobility services such as Uber, Lyft, and Didi are highly dependent on the understanding of individual travel demand.

Furthermore, comparing to answer the questions about current situations, a more challenging task transportation planners are usually facing to is forecasting the future demand. In recent years, population overcrowding happens in lots of countries especially in urban area. which cause a series of problems and increase the burden for current public transport system, road networks and infrastructures. Considering the limited public space in urban area, it is not practical to construct more transport facilities. On the contrary, depopulation occurred in rural area makes local government have to roads and bridges because of the drop of users. In summary, how to effectively utilize current resources and correctly capture the changing population travel demand is the key factor to solve next generation urban problems.

To achieve these goals, the requirements for understanding people movement on citywide level are manifolds.

- The knowledge of people movement must be up-to-date that is capable of replicating current situations on time.
- The granularity of the replication of must be proper to reveal both decision makings on individual level and phenomenon on population level.
- The forecasting for future or unprecedented scenarios must be correct to support decision making and planning.

TABLE 1.1: Comparison between most existing methods for people flow monitoring and estimation

	Macro	Micro	No Delay	Forecasting
Traffic Census	✓	✗	✗	✗
Travel Behavior Survey	▲	✓	✗	✗
Travel Demand Estimation	✓	✓	✗	✓
Urban Monitoring	✗	✗	✓	✓
Human Mobility Modeling	✗	✓	✓	✓
Objective Approach	✓	✓	✓	✓

✓ = provides property; ▲ = partially provides property; ✗ = does not provide property;

However, most of existing approach could not satisfy all the requirements discussed above. As shown in Table 1, we list existing approaches about the topic of "understanding people movement on citywide level" from different domains including transportation, socioeconomic, geography and computer science. The most straightforward method is traffic census that using tally counters. Though the result from census is reliable to capturing the overall travel demand on target area, the high cost of hiring large amount of crews and processing time make it to be impractical to conduct the census frequently. Besides, the results can only provide the amount of traffic volume, lack of detailed trip information and traveler background limits its application for future forecasting and decision making support.

1.1.1 Survey-based approach

On the other hand, travel-related surveys are conducted to study individual travel behavior. Generally, subject's trips (origins and destinations, start time and end time, transport mode and purpose) on a given day are recorded with individual attributes such as socio-economic and demographic information are collected. Based on these data, critical important knowledge such as current traffic volume estimation in the form of origin-destination matrix [62, 15], future travel demand forecasting and individual travel preferences are produced to support decision makers for future plan. However, such manually collected questionnaires are extremely expensive and cost long time for data processing. Thus, the survey cannot be conducted frequently and the results are usually suffering significant delays. Obviously, thus methods cannot meet the current challenges.

1.1.2 Travel demand modeling approach

During last few decades, researchers have made significant progresses in modeling and estimating travel demand. Generally, travel demand models are developed to forecast the response of transportation demand to changes in the attributes of people using the transportation system. Specifically, travel demand models are used to predict travel characteristics and usage of transport services under alternative socio-economic scenarios, and for alternative transport service and land-use configurations[11]. Previously, travel demand approach uses trips as the basic unit of modeling. Four separate steps of procedure are developed for estimating the total inflow and outflow of each zone in the target area (as ‘ trip generation ’), assigning trips to each zone pair (as ‘ trip distribution ’), determining transport mode of each trip (‘ as mode choice ’) and finally assigning trips to road network (‘ trip assignment ’). However, trip based travel demand approaches are failed to modeling individual daily schedule with trip chain structure because all the trips generated from the models are separated and there is no behavior rule to combine them in rational order. To fill this gap, activity-based travel demand approach is developed. Thus approach is on the basis of that the travel demand is the result of participating activities at different places and time. Since activity is more easily to be modeled and forecast at individual level, it immediately replaced traditional trip-based approaches. Travel behaviors are regarded as the derivatives of activities at different places such as home, work, shopping and others. Especially, the discrete choice models are widely used for modeling activity choice, departure time choice, transport model and other behavioral factors. However, developing activity-based models needs detailed travel behavior survey, the data collection is usually expensive and with significant delay. Because of this limitation, only typical day’s travel demand can be modeled and estimated which obviously match the current requirements from demand-side. Furthermore, few studies have examined the spatial-temporal aspects of travel choice simultaneously, spatial choices are often on zone level and temporal aspects are ignored. Resolution on microscopic level are not enough.

1.1.3 Emerging data collection and urban monitoring approach

With the development of Internet of Things (IoT) and Information and Communication Technology (ICT), individual travel footprints can be sensed and recorded by more and more services and devices. The most well used data source of the population, Call Detailed Record (CDR) is collected by mobile phone carriers. The record is generated when telephone call, text message or Internet data exchange that passes through that devices. The device carrier's location is recorded by the nearest base tower number. The spatial resolution of such records varies from several hundreds meters in the central of urban area to few kilometers in rural area. According to its high population coverage, thus data are well studied and leveraged for human mobility pattern analysis[21, 54, 25], link traffic volume estimation [60] and crowd density monitoring[14, 36]. On the other hand, due to the popularization of the smart phones, GPS data is also widely used and collected from location based services such as navigation, check-ins, recommendation, disaster alerts and advertising. Comparing to CDR data, spatial resolution of GPS is less than 10 meter in most devices which enables researchers to mining more detailed information such as carrier's travel speed, transportation mode, points of interests. However, current applications of location big data are still limited in data mining and aggregated level, even though the results can be correctly estimated, it is still hardly to figure out how individuals move over time and unable to capture where the people flow come from and where they will go to.

Human mobility modeling

On the other hand, researchers from different domains start to using CDR and GPS data to reveal the nature and pattern of human mobility. The beginning of this topic is started by [21, 54], which reveal statistical characteristics and predictability of individual mobility by mining CDRs data, [49] uses same data and summarize human daily mobility pattern into 17 unique types. The objectives of these studies reveal multiple dimension of human mobility such as destination, departure time choice, travel distance, transport mode classification, trajectory matching, activity choice and activity pattern. However, most studies just focus on a single aspect of human mobility, few studies integrate these separate factors to modeling daily schedule.

1.1.4 Machine learning approach

In recent years, benefiting from the explosive development of deep neural networks(DNNs), machine learning has been shown to be the most exciting approach and has achieved a lot of success in natural language process, objective detection and computer vision. The complex and non-linear human mobility can be represented by DNNs even without advanced domain knowledge. For example, state-of-the-art recurrent neural network[73] has been successfully applied in modeling and predicting sequential behaviors, in transportation domain, [35] extended the model for next place prediction by adding spatial and temporal contexts. [70] leveraged Long Short-term

Memory(LSTM) to model individual's daily activity schedule, thus generative model is also able to reproduce synthetic activities. On aggregated level, [71, 72] proposed citywide crowd flow prediction method using deep residual network. Comparing to traditional approaches, machine learning models are accomplished in handling large amount features and complex pattern representation. However, for most studies, the modeling and forecasting are on the basis of daily trajectory analysis, thus the results are hard to applied into whole population and unprecedented scenarios (where training data is hard to get). Besides, training robust machine learning (especially deep learning) model needs large amount training data which only few privacy companies collect, store, and manage them. Due to the strict privacy policy constraints, although data collected from who has agreed to provide their location information is allowed to be used for research purposes, the data is managed anonymously and it is difficult to associate demographic and socio-economics attributes with travel behaviors. Because of this limitation, few of machine learning approaches studied the decision-making of travel behavior.

Reinforcement learning is another branch of machine learning that integrates with agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Unlike machine learning that needs to learn from a training data set and then apply the trained model to new data set, reinforcement learning is dynamically learning by adjusting actions based on the feedback from interaction with environment. Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. The agents generate different episodes(sequential of actions) and learn from the feedback whether that lead to a good result, then reinforce the actions that worked, otherwise lower down the probability of choosing these actions. Comparing to traditional trip-based and activity-based travel demand modeling approach, reinforcement learning could handle more complex environment with plentiful algorithms. It also enables researchers to combine multiple aspects of human mobility such as decision choice, time choice and transport mode.

Reinforcement learning has already been applied to modeling travel demand related decision making[24, 68]. However, all of these formulations use a predetermined reward function, a significant limitation to the use of " forward " RL methods alone to learn decision rules for activity scheduling[17]. Although inverse reinforcement learning[42] has been proposed to solve complex multidimensional reward function formulation issue, there is no existing work combines IRL approach with travel demand modeling.

1.2 Objectives and Originality

In this thesis, we aim to develop a novel reinforcement learning based agent modeling and simulation framework that is capable of revealing both individual level daily travel behavior decision-making and citywide level people flow dynamics for various scenarios. To achieve this goal, we extend the current reinforcement learning framework to real world travel demand modeling problem by multiple steps. First, we designed the urban environment model which provides sufficient and accurate state-action space that efficiently decrease the calculation cost. Then we discuss the algorithms for finding optimal policy. To have a robust and fast training process, we explored several value-based and policy based algorithms including tabular solution methods and approximation methods with the power of neural network. Another challenge is how to derive human behavior preferences from passively collected location data. We introduce inverse reinforcement learning techniques that are capable of recovering reward functions from demonstration trajectories. To achieve this goal, we first proposed a method to process raw locational data into individual's daily trip with transport mode from raw GPS data. The preprocessing pipeline overcome the data sparsity issue and can be applied to other mobility datasets. By inferring and implementing the behavior preference parameters to agent model, the agents are capable of of learning mobility sequences from raw locational data while incorporating behavioral parameters that are sensitive to environmental context is introduced.

We applied the models to the data collected from GPS data collected by a smartphone application and People Flow Data (a data processed from Person Trip Survey which consists of spatio-temporal information) in urban area in Japan. The agent models are validated and compared with existing behavior models. The results show that proposed model framework outperforms the straightforward reinforcement learning, with the power of learning from demonstration data. On the other hand, because the high flexibility of agent models, it is difficult to evaluate the generate trajectories. To better evaluate the proposed modeling framework, we simulate daily people movement based on the developed model and compared the simulation results with ground truth. Experiments are both launched on normal day and disaster scenarios.

Another important application of the agent-based modeling and simulation framework is to forecast people movement in unprecedented scenario. Lack of historical data is the major challenge for direct forecasting because the collection and storage of emerging data sets are just started from recent years. We focus on the case that the same event (i.e. disasters) have happened at other places, where the data of people movement is sensed and stored. We use the mobility data collected from other places and develop the agent models, then simulate people movement in target area.

The originality of this thesis is summarized as follows;

- First, we proposed an reinforcement learning based agent model that is capable of self-teaching and learning from trial and error. Given different environment settings, agents can autonomously find out optimal policy.
- Second, with the proposed reinforcement learning framework, we integrate multiple aspects of human mobility such as destination choice, transport mode, departure time choice and behavior pattern in daily scheduling issue to generate realistic individual travel behavior. Unlike traditional activity-based modeling approach, our approach is a straightforward tour-based travel demand model, which could simultaneously reveal spatial and temporal choice with higher result resolution.
- Third, we first introduce inverse reinforcement learning into travel demand modeling approach to improve agents performance. IRL can be seen as a method to imitate how real people behave from their trajectories. By sampling and learning from the large amount mobility data set, the reinforcement learning agents can approximate the behavior of the real people who contributed their data to the dataset.
- Finally, based on the proposed agent model, we develop an agent-based simulation framework that provides a bottom-up travel demand estimation replication which could support decision makers from multiple perspectives.

1.3 Overview

The overview of this thesis is structured as follows:

- In chapter 2, we review most previous studies related to this thesis, including travel demand modeling and simulation approach from different approaches, human mobility modeling and machine learning approaches by leveraging large scale anonymous location data on both individual and aggregated levels.
- Chapter 3 introduces our reinforcement learning based agent models from anonymous location data. We present the modeling framework, agent and environment formulation, and explain the algorithm in both small and large state space settings.
- Chapter 4 describes the inference method of recovering human travel preferences from anonymous location data using inverse reinforcement learning techniques. To do so, we explore the processing pipeline from raw sparse GPS data to learn-able demonstration trajectories, and applied Maximum Entropy Inverse Reinforcement Learning algorithm to the RL framework proposed in Chapter 3. We also discuss the relationship between the trajectory data and estimation results of inverse reinforcement learning with different volumes and on different areas.
- In chapter 5, we develop the application of people flow on citywide level in different places and scenario using the agent models in Chapter 3 and 4. We propose a modeling and simulation framework for travel demand forecasting problem that combine emerging anounymous data with promising agent-based techniques. The agent models are evaluated in different levels and scenarios. The results show that proposed

agent model can well reproduce people daily travel planing problem on different areas and situations. Furthermore, an application of transferring pre-trained agent model to new environments are also be validated.

- Finally, we summarize the research motivation, objective, evaluation of experiments results, and simulation applications in Chapter 6. Future research directions for more comprehensive applications are also discussed here. At last, we summarize the current limitations and future directions.

1.4 Contribution

In this dissertation, we aim to develop an reinforcement learning based agent model for reconstructing people flow on citywide level. We first review the current challenges of understanding mass people movement on citywide level and existing monitoring and modeling approaches. Then we combined current agent-based simulation framework with reinforcement learning techniques in space movement issue for real-world applications and evaluated the results. The contribution of this study are summarized as follows.

- We proposed and end-to-end data processing and modeling pipeline to integrate reinforcement learning and location data. Location data such as GPS data, person trip survey data were used as input, and the framework can be used to create detailed temporal travel behavior profiles. By this processing pipeline, data sparsity issue is overcame.
- Combining this with an RL approach especially for the objective of generating synthetic human mobility traces from anonymous location data

makes it possible to expand the scope of the state-of-the-art ABM techniques for travel demand analysis.

- We simulated people flow in different area and scenarios to evaluate model performance with real world dataset.

2 Related works

In this chapter, we summarize the most existing methods about capturing people movement with different approaches such as questionnaire survey, travel demand modeling, emerging data collection and analysis and machine learning approach.

2.1 Survey-based Approaches

Traditionally, travel behavior surveys are widely used by transpiration planners for decision making, system design and policy evaluation. The surveys are commonly in a diary format, which first appeared in the late 1970s in German, and sooner be introduced to the United States [51]. Participants are required to report their typical travel behaviors in a common day with the details of each trip including origin, destination, transport mode and travel purpose. Participants' personal information such as age, gender, occupation and ownership of private cars are also collected. Information is collected manually by the means of paper questionnaire, telephone and face-to-face interview. More details of travel survey history and development can be found in [57, 45]. Take Japan as example, the Person Trip (PT) survey has been conducted in around 62 cities with sample sizes that were as big as 1-3 percent[1]. Besides, another approaches named tracking method is also used to collect travel movement of sampled individuals [6]. Although tracking surveys can provide more details about space-time movement of target individual, the high cost for for tracking people movement in urban scope limits its application in large samples.Based on the survey results, transportation researchers can easily summarize characteristics such as average trips, the transport modal share, travel time and distribution of trip generation. The present origin-destination matrices can also be estimated on Traffic Analysis Zones level as an important production of survey.

Travel survey provide the foundation to reveal current travel situation. On the contrary, travel demand models are developed to forecast future demand. There are two types of travel demand models, Four Step Model (FSM) and Activity based Model (ABM). The FSM is consist of trip generation, trip distribution, modal split and traffic assignment. In the first step, the total number of trips from a particular traffic zone is estimated based on the demographic and socio-economic characteristics on the basis of census and travel survey results. Then, trips are distributed from generated zones to attracted zones. In the third step, mode split estimates the share of trips between each generated zone and attracted zone that uses a particular transport mode. Finally, all of the distributed trips are assigned to the existing transportation system. Although this approach has achieve success in aggregated level estimation and forecast, it has failed to perform in most relevant policy test, whether on the demand or supply side [38].

The FSM can also be seen as a type of trip-based model which each trip is generated separately. To improve the consistency of travel representation, researchers [22, 13] group trips into tours on the fact of all travels can be regarded as round-trip based at home. The shift from trip to tour based enable model system to incorporate with temporal-spatial factors.

2.2 Travel demand modeling approach

During last few decades, researchers have made significant progresses in modeling and estimating travel demand. Generally, travel demand models are developed to forecast the response of transportation demand to changes in the attributes of people using the transportation system. Specifically, travel demand models are used to predict travel characteristics and usage of transport services under alternative socio-economic scenarios, and for alternative transport service and land-use configurations[11]. Previously, travel demand approach uses trips as the basic unit of modeling. Four separate steps of procedure are developed for estimating the total inflow and outflow of each zone in the target area (as ‘ trip generation ’), assigning trips to each zone pair (as ‘ trip distribution ’), determining transport mode of each trip (‘ as mode choice ’) and finally assigning trips to road network (‘ trip assignment ’). However, trip based travel demand approaches are failed to modeling individual daily schedule with trip chain structure because all the trips generated from the models are separated and there is no behavior rule to combine them in rational order.

Activity-based models are the most promising modeling approach that could reveal individual’s daily behaviors[9]. The theory of activity based model is on the idea that the demand for travel is derived from the demand for activities[13, 27]. The model incorporates demographic factors (age, gender,

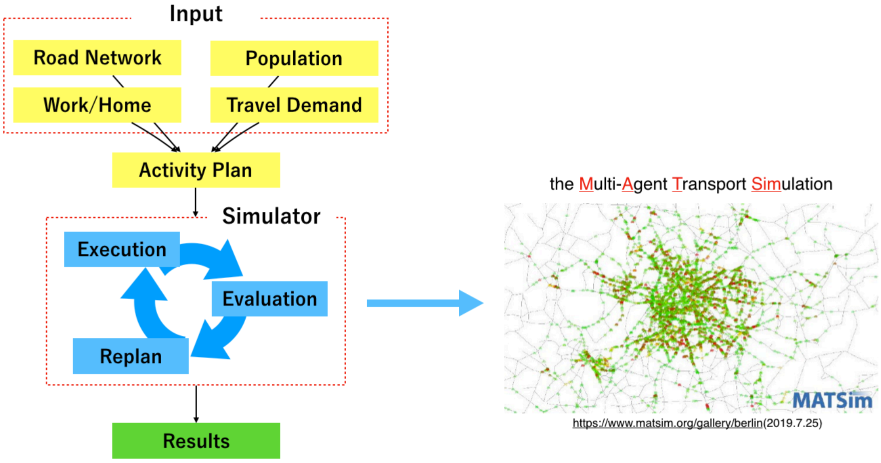

FIGURE 2.1: An example of activity-based modeling and simulation framework

occupation and household members) with the choice of activities[43], dependence of destination choice in trip chains[28], and activity duration models[10]. [13] combines all these highlights and proposed a multi-level nested-logit choice model that successfully represent daily activity decision makings. Thus model is widely used for travel demand modeling, traffic simulation and future demand forecast. We give a example framework in Fig.2.1. Comparing to previous modeling approaches, activity based model is capable of expanding to full population considering demographic attributes, sensitive to environment and policy changing and flexible to combine with temporal-spatial constraints, thus it is widely used for evaluating how policies, urban changes and rare events will affect people travel behaviors.

TABLE 2.1: Summary of existing travel demand estimation approaches

	Method	Author	Source	Objective	Detail
Trip based approach	Four step model	Sasaki et al. (1972) [48]	Land Use	Trip-chain	Markov chain
		Golob et al. (1986) [20]	Land Use	Trip-chain	
Activity based approach	Discrete Choice Model	Ben-Akiva et al. (1994)[12]	Time use Survey	Daily plan	Static
		Ben-Akiva et al. (2007)[13]	Time use Survey	Activity choice	Dynamic
		Wen et al. (2000)[65]	Time use Survey	Car ownership	
	Hazard Duration Model	Hamed et al. (1993)[23]	Time use Survey	Activity Duration	
		Vause et al. (1997)	Time use Survey		
		Gnarling et al. (1989)	Time use Survey	Departure time	
	Rule based Model	Pendyala et al. (1995)[44]	Time use Survey	Activity chain	
		Arentze et al. (2008)[5]	Time use	Activity chain	

On the contrary, agent-based Modeling (ABM) is a powerful tool for studying self-organizing system with heterogeneous agents situated in a shared environment. In the last few decades, ABM has been applied to lots of domains include social sciences, signal control and transportation simulation. In these researches, agents may correspond to cities, blocks, platoons, households, individual travellers (drivers), vehicles, sensors, traffic signals, etc [7]. However, it is often difficult to develop a reliable agent-based model since one needs to develop agents ' behavioral rules through a qualitative understanding of the domain and careful calibration of agent and environmental parameters [30]. Moreover, traditional ABM practice relies heavily on expert opinion or qualitative comparisons of behavior to develop robust model. To overcome these issues, recent years researchers attempt to incorporate machine learning techniques and realistic datasets into ABM methods. However, most of these methods are black box machine learning models; the open challenge is how to interpret the results, which hinder the applications of these methods in ABM.

2.2.1 Emerging data collection and human mobility modeling approach

With the development of Internet of Things (IoT) and Information and Communication Technology (ICT), individual travel footprints can be sensed and recorded by more and more services and devices. The most well used data source of the population, Call Detailed Record (CDR) is collected by mobile phone carriers. The record is generated when telephone call, text message or Internet data exchange that passes through that devices. The device carrier's location is recorded by the nearest base tower number. The spatial resolution of such records varies from several hundreds meters in the central of urban area to few kilometers in rural area. According to its high population coverage, thus data are well studied and leveraged for human mobility pattern analysis[21, 54, 25], link traffic volume estimation [60] and crowd density monitoring[14, 36]. On the other hand, due to the popularization of the smart phones, GPS data is also widely used and collected from location based services such as navigation, check-ins, recommendation, disaster alerts and advertising. Comparing to CDR data, spatial resolution of GPS is less than 10 meter in most devices which enables researchers to mining more detailed information such as carrier's travel speed, transportation mode, points of interests. However, current applications of location big data are still limited in data mining and aggregated level, even though the results can be correctly estimated, it is still hardly to figure out how individuals move over time and unable to capture where the people flow come from and where they will go to.

2.3 Machine Learning based Approach

With the rapid development of ubiquitous technologies, people's movements are sensed by various approaches such as Call Record Details (CDRs), credit card bills, GPS-equipped devices, social network check-ins and public transit smart cards. Among them, CDRs, GPS and social networks data are the most popular data source for studying people's travel behaviors. CDR data usually has the highest coverage including millions of users, although the temporal and spatial resolution are rough, recent studies have developed efficient pipeline to detect user's activities and behavior pattern[26]. On the contrary, GPS data has highest spatial accuracy, and can be easily collected by numerous kinds of location based services, which automatically report devices holder's location in short time interval[67] (or when applications are using). However, both these two kinds of data are under strict privacy policy, the personal information is hard to incorporate with behavior analysis and only the services provider can use the data for research purposes. Besides, social network data such as check-ins, location-based tweets is also important data source for researchers. Although the consistency of data is lower than CDR and GPS, the openness to the public and richness of traveler's background, comments on travels open the door to mining further knowledge about travel behaviors.

Studies on emerging data sets can be divided into two types as individual based approaches and population based approaches. Individual based approaches are focusing on the prediction of future behaviors. Since GPS and CDRs is capable of observing individuals' travel behaviors in long time period, the regulation of daily travel behavior makes it possible to predict individual's location transition. Classical methods such as simple Markov model[19], Hidden Markov model[37] have been developed for next places prediction. [16, 69] achieve context-aware by combining with environment information. However, the prediction accuracy is dependent on the amount and quality of observation data. Besides, the learned models cannot be applied in new environments or scenarios. Previous activity based methods have also been applied on CDR data[26] for individual based travel behavior study. The basic idea is to infer activities from mobility data and use the inference results for activity based modeling. Although such approaches make up for the out-of-data issues of traditional data sets, without additional information only few activity types can be recognized (i.e. home, work, school and others), and lack of individual background information also limits the applications. Recently, machine learning models have also been introduced to this domain. Graph-based behavior inferences[56], transportation mode detection[75], trajectory prediction using LSTM[4] are applied to location big data with the power of neural networks.

On the other hand, applying location data on aggregated level does not suffer the constraints of privacy policy issues, so the applications on such domain are more well developed. The beginning of this topic is started by [21, 54], which reveal statistical characteristics and predictability of individual mobility by mining CDRs data. Following studies[49] uses same data and summarize human daily mobility pattern into 17 unique types. The collection of large amount mobility data brings convenience for sensing and modeling mass people movement in city-scale. The first work in this domain is [14] that used network usage to detect crowd dynamics in a particular area. Inspired by this research, a series of studies attempted to detect population movement status by using CDRs [8]. Once home locations are detected[], the population of observed mobile phone users can be expanded to whole population level by incorporating with National Census. Another promising direction of applications are "urban monitoring" that treat mobile device holder as sensor to detect crowds congestion and population distribution.

2.4 Reinforcement Learning

Reinforcement Learning (RL) is an area of machine learning that focus on the interaction between agents and environment, to derive optimal policy by trial and error based on sequential decision-making process which can be applied in a variety of fields such as robot control, video games and system optimization [39, 47]. The theory of RL provides interpretable, psychological and neuron-scientific perspectives on human behavior, of how they plan their actions in a given environment[58]. The framework of reinforcement learning provides a mathematical formalization of intelligent decision making that is powerful and broadly applicable for agent control [31, 34]. However, for a long period, their applications are limited to domains in which agent behave in low-dimensional state spaces with well-defined reward function.

Over the past few years, RL has become increasingly popular due to its success in addressing challenging sequential decision-making problems. Several of these achievements are due to the combination of RL with deep learning techniques[40, 39, 61]. This combination, called deep RL, is most useful in problems with high dimensional state-space. Previous RL approaches had a difficult design issue in the choice of features. However, deep RL has been successful in complicated tasks by using less prior knowledge thanks to its ability to learn different levels of abstractions from data [**I**]. For instance, a deep RL agent can successfully learn from visual perceptual inputs made up of thousands of pixels. This opens up the possibility to mimic some human problem solving capabilities, even in high-dimensional space which, only a few years ago, was difficult to conceive.

However, although reinforcement learning approach has proved its potential in lots of domains, few real world challenging tasks are solved by reinforcement learning. There maybe few reasons for this issue. First, traditional reinforcement learning algorithms such as dynamics programming, value iteration are capable of solving precise policies for small scale state space RL problems, but real world applications usually have a enormous state space that have to be calculated by approximate approaches with the power of deep learning, that is hardly to achieve real time solution. Second, the representation of real world environment is too complex and there is still no common solutions. Third, the reward of RL problems are usually clear and sparse (like win a game, got a item or achieve some beforehand goals). However, the reward of real world applications, especially for human being tasks, the reward function are hard to define. As consequences, reinforcement learning agents are hard to behave like real human-beings.

Several notable works using deep RL in games have stood out for attaining super-human level in playing Atari games from the pixels[39], mastering Go [53] or beating the world ' s top professionals at the game of Poker. Deep RL also has potential for real-world applications such as robotics [33], self-driving cars[**pan**]and smart grids. Nonetheless, several challenges arise in applying deep RL algorithms. Among others, exploring the environment efficiently or being able to generalize a good behavior in a slightly different context are not straightforward. Thus, a large array of algorithms have been proposed for the deep RL framework, depending on a variety of settings of the sequential decision-making tasks.

Another promising method is inverse reinforcement learning. Inverse Reinforcement Learning (IRL) enables robots to learn complex behavior from human demonstrations, in cognition and preference learning, where it serves as a tool to discover human behavior preferences. The objective of IRL is to estimate the reward function from experts' trajectories that motivates agents' behavior underlying the environment. By recovering the reward function correctly, agents are capable of imitating the experts' behavior. Lots of the previous works in this domain relies on parametrization of the reward function on the basis of hand crafted features[32, 63]. Furthermore, this approach makes the transfer of well-trained reward functions between different scenarios under the same feature representation and achieve better generalization performance than direct state-to-state mapping to be possible [77, 42, 2]. Especially, previous studies represent the reward function as a linear function with hand crafted features. In this study, we expand the use of such a framework on recovering the preference of daily movement decision making based on reward function and an attempt is made to replicate citywide level population ’ s flow by incorporating with agent-based multi-modal traffic simulation technologies.

3 Modeling Travel Behavior as Reinforcement Learning

3.1 Introduction

As mentioned in Chapter 2, most of reinforcement learning (RL) studies are only for toy domains but seldom to solve challenging real world problems. In this chapter, we extend the framework of modeling and simulating individual daily travel behavior as reinforcement learning problem. The reinforcement learning models reveal subject's behaviors in the form of sequential decision makings by Markov Decision Processes (MDPs). We focus on the behavior of travel behavior considering of destination and transportation mode choices. The model can be considered as a type of trip-based model, and we do not consider specific activities or trip purposes.

3.2 Modeling Framework

In this study, we aim at modeling and simulate human daily travel behavior based on RL framework. Although RL enables agents to learn in an interactive environment by trial and error using feedback from their own

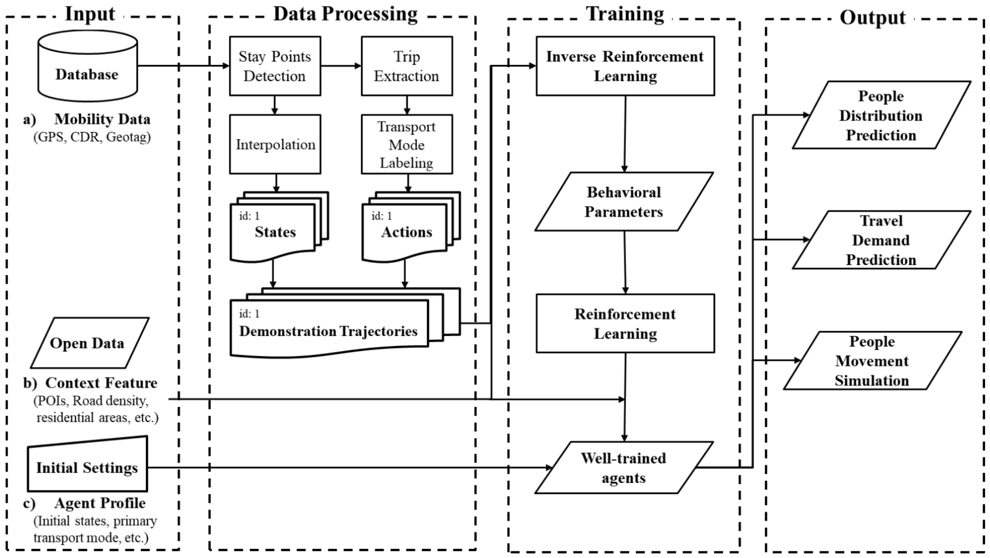

FIGURE 3.1: Framework of this study

experience, there is not enough knowledge to design a perfect reward function whose optimization would generate human-like behavior, so straightforward RL may not lead to a realistic simulation result. One solution is to mimic the behavior of human-beings and characterize the set of reward functions. Thus, human behavior observations are needed as training data. As shown in Fig. 1, the end-to-end framework consists of three parts: developed data processing, agent modeling with parameter training, and agent-based travel micro-simulation.

3.3 Formulation of Agent Model

Markov decision processes (MDPs) provide a mathematical framework for modeling the sequential decision-making process in a variety of situations where the outcomes are under the control of a decision maker[58]. The agent

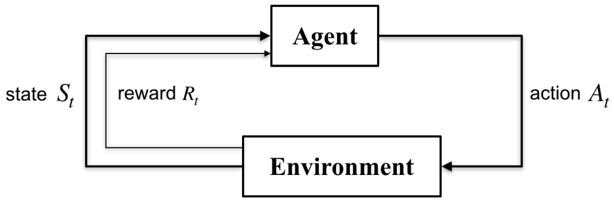

FIGURE 3.2: The interaction between agent and environment in the MDP framework.

choose actions and the environment returns feedback to agent and transit agent to the next state. Fig depicts this mechanism. Thus, we represent individual daily travel behavior decision making as a deterministic MDP. The individual traveler is modeled as the agent. Everything out of the agent where it interacts with is modeled as environment. The agent selecting actions and the The agent observes its state at every time step from the environment and takes action accordingly. Then the agent transits to the next state and receive reward from environment.

Generally, a MDP is a tuple of five elements (S, A, T, R, γ), where

- State: S is the state space that an agent can visit. A state s_t is defined as the location of the person's stay at a specific time step t. Further, this space is expanded into higher dimensions by functions mapped from s_t into $\mathbb{R}$ which denoted as: $s = [lon, lat, t, x_1, x_2, \cdots, x_n]$. The triplet $[lon, lat, t]$ represents the individual's location and timestamp, and $[x_1, x_2, \cdots, x_n]$ is a set of context features consisting of numerical data related to a set of context information.

- Action: A is a finite set $a_1, a_2, \cdots, a_M$ of actions. An action $a \in A$ is denoted as $a = [destination, mode]$ which controls agent's transition between states. The agent interacts with environment at each time steps as $t = 0, 1, 2, 3, \cdots, $. At each time step t, the agent observes its current state $S_t \in S$ from environment, and chooses an action $a_t \in A(s)$ based on current state. To reduce action space and computational cost, the environment is simplified by splitting space into a 1-km grid as the transition unit. Furthermore, the structure of the road and railway network constrained the accessibility for some areas (i.e., some rural areas are not reachable by railway), making it necessary to filter out these kind of actions. The subset action of state s, $A_s \in A$ to improve performance.

- Transition function: The transition function is the system dynamics. Basically, this function it is a probability distribution over next possible successor states, given current state and action as $T = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ that action a in state s at time t will lead agent to state s' at next time step. In this study, because the definition of action has figured out specific destination which represents the next state, we set transition probability as 1.

- Reward function: The reward function is indispensable for computing an agent ' s policy $\pi(a|s)$. An appropriate representation of reward leads agent to generate desirable behavior. However, a " desirable behavior " is extremely hard to define and varies from person to person. Therefore, in this study, an multi-attribute reward function $R(s, a; \theta)$ is proposed that defines the immediate reward of action a and state s. The feature vector f is mapped from state-action pair (s, a), which incorporate the information from the current state, destination, and travel cost from action a considering the transport mode. In this study, R is assumed to be the linear combination between the features $f(s,a)$ and a weight θ. From this perspective, the parameter θ_i corresponding to feature x_i represents the preference of taking such an action.

- Discount factor: $\gamma \in [0,1]$ represents the weight for a step of from a state steps into the future. This factor is usually to use for control episode length.

In a RL problem, the agent tries to maximize the cumulative reward it could receive in future. To be noticed, the maximization is not focus on one-step (or next step) but cumulative reward in the future considering the discount factor as follows:

$$R_t = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots] = E[\sum_{k=0} \gamma^k_{t+k}] \tag{3.1}$$

3.4 Reinforcement Learning Algorithm

The obvious way of finding an optimal behavior in some MDP is to list all behaviors and then identify the ones that give the highest possible value for each initial state. Since, in general, there are too many behaviors, this plan is not viable [59]. Formally, a policy is the action selection denoted as probabilities of selecting each possible action under a given state. If the agent follows policy π at time t, then $\pi(a|s)$ is the probability that $A_t = a$ if $S_t = s$. The objective of RL is to find an optimal policy that leads agent to collect maximize the discounted reward over time.

In this section, we discuss reinforcement learning algorithms to let agent to learn proper policy to behave. In figure 3.2, we list most of current algorithms

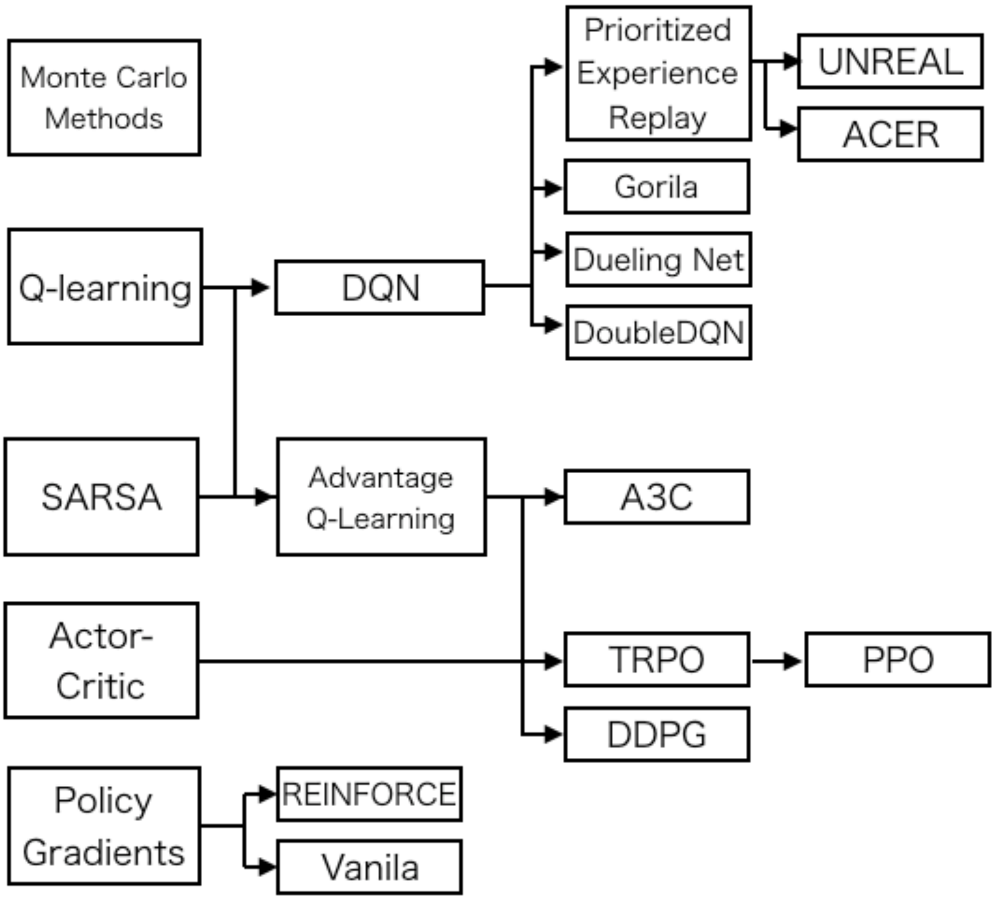

FIGURE 3.3: Current Reinforcement Learning Algorithms

that are developed to solve reinforcement learning problems in different scenarios. Generally, the algorithms can be divided into two categories: the tabular solution methods and approximation approaches. The former tabular solution methods focus on the cases in which the state and action spaces are small enough so that the state value and action value (will be explained later) can be stored as array or table for programming. The fundamental classes of methods for solving finite Markov decision problems include dynamic programming, value iteration, TD learning and so on. In the section 3.3.1, we leverage tabular solution methods to solve RL agent policy in small state spaces and in section 3.3.2, we introduce deep reinforcement learning algorithms to apply agent model into high-dimension large state-space for real world applications.

3.4.1 Algorithms for Reinforcement Learning in Small State Space

As aforementioned, tabular solution methods are able to find exactly the optimal policy. For infinite MDPs, the optimal policy is always better than (or at least equal to) other policies for all states. We fist discuss the algorithms that is capable of solving optimal policy and apply the algorithm to travel behavior agents.

The key idea of finding optimal policy in reinforcement learning is to estimate the value functions of state-action pairs that evaluate the performance of choosing an action in a given state. The performance can be calculated in terms of expected feedback from environment. Accordingly, value functions are defined on the basis of policy. The value of a state s under policy π is represented as $v_\pi(s)$ that represent expected feedback from environment when starting from state s under policy π. For MDPs, $v_(\pi)$ can be formulated as follow:

$$v(s) \doteq \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^t R_{t+k+1}|S_t = s], foralls \in S \tag{3.2}$$

where $\mathbb{E}_\pi[\cdot]$ denotes the expected value of a random variable given that the agent follows policy π, and t is any time step. Similarly, the value of choosing an action a in state s under a policy π, written as $q_\pi(s,a)$, as the action value can be defined as follow:

$$q_\pi \doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a] \tag{3.3}$$

The optimal policy π_* may be not unique, all the optimal shares the same optimal state-value function v_* defined as follow:

$$v_*(s) \doteq \max_\pi v_\pi(s), \tag{3.4}$$

for all $s \in S$. Similarly, the action-value function Q_* can also be denoted as:

$$Q_*(s,a) = \doteq \max_\pi q_\pi(s,a), \tag{3.5}$$

for all $s \in S$ and $a \in A(s)$. The relationship between state-value and action-value can be defined as:

$$Q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t = s, A_t = a] \tag{3.6}$$

The relationship between the value of a state on the basis of an optimal policy and the expected feedback for the optimal action choice from that state can be expressed by the Bellman optimallity equation as:

$$\begin{aligned} v_*(s) &= \max_{a \in A(s)} q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}[G_t | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')] \end{aligned} \tag{3.7}$$

The Bellman optimality equation for q_* is

$$\begin{aligned} q_*(s,a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1},a') | S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma \max_{a'} q_*(s',a')] \end{aligned} \tag{3.8}$$

There are three fundamental classes of methods for finding the optimal policy π_*: dynamic programming, Monte Carlo methods, and temporal-difference learning. We implemented dynamic programming method as the basic solution of reinforcement learning of this research because it provides the most complete mathematical theory to achieve the optimal policy. In dynamic programming, value functions are updated by value iteration algorithms.

Algorithm 1: Value Iteration algorithm for estimation π_*

```
Result: Output a deterministic policy, $\pi_*$ such that
        $\pi(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$
Set a small threshold $\theta > 0$ determining accuracy of estimation;
initialization;
while $\Delta < \theta$ do
    $\Delta \leftarrow 0$;
    for $s \in S$ do
        $v \leftarrow V(s)$;
        $V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$;
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$ ;
    end
end
```

The major limitation of the value iteration method is that it involves operations over the entire state set of the MDP. If the state set is very large, the calculation will be extremely expensive. Compared to normal MDP problems, urban area is a more complex system that could generate an extremely large state space that is not capable of being computed. To avoid this problem, the key idea is to reduce the size of state space and preparing a necessary and sufficient action set to make the problem tractable. As explained in Section 3.2, by discretizing space into mesh grid can efficiently decrease the size of state space, however, the granularity of discrete space is a trade-off between calculation cost and prediction accuracy. We found that population density is very high in urban area, there are still lots of areas which are unreachable for human mobility. The accessibility of each places determined by its land use,

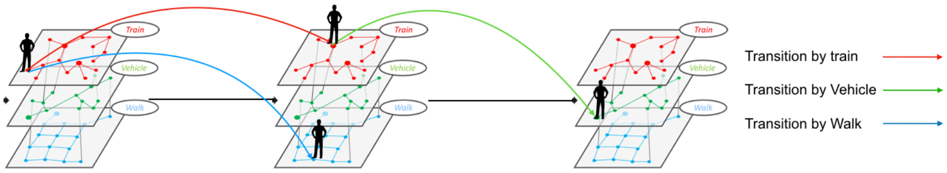

FIGURE 3.4: Graph based state-action representation of urban environment

transportation infrastructure and status. For example, a place (or a grid in the discrete space) without railway station will not be potential destination of a "railway" trip. Although it is impractical to determine all these relationship between difference places, we can connect geographical areas and create the transition relationship by mining the human daily movement trajectories.

Definition 1 *Given two mesh $m(x,y)$ and $m'(x',y')$ in target area, if a trip starts from m and arrives at m' (and vice-versa), the m and m' are connected. Furthermore, if the trip with specific transport mode (i.e. vehicle, railway or walk), we call the two meshes are connected by the mode.*

Figure 3.3 shows the image of location connected relationship. To further suppress the state, we introduce a graph-based representation to depict urban areas to further reduce the size of state and action set.

Definition 2 *A graph G is consist of a finite set of nodes V and edges E, we denote $v \in V$ as a specific connected location following the definition 1, and $e \in E$ as a trip connect two locations with a particular transport mode. To apply this graph-based representation to MDP, we define states $s \in V$ and actions $a \in E$. Thus, $|s| = |V|$ and $|A = |E|$. The edges connected to state s denoted as $E(s)$ is the subset of action space $A(s)$ which direct the potential destination from state s.*

3.4.2 Algorithms for Reinforcement Learning in Large State Space

In many of the tasks in transportation domain, the state space is complex and enormous, such as higher level mesh system (i.e. discretize space at 500m or 100m) and road networks which have more than 100,000 nodes and links in citywide level. In such cases, we cannot expect to find an optimal policy or the optimal value function even in the limit of infinite time and data; our goal instead is to find a good approximate solution using limited computation resources. The calculation time and memory needed are not impractical. Besides, the graph-based representation of state-action space described in 3.3.1 is highly dependent on existing mobility data, without the help of data, it is hard to design a robust state-action space and delicate model for reinforcement learning. To overcome this issue, one promising one is to use neural networks to represent state value function $V(s)$ and the action-value function $Q(s, a)$ [41] for approximation. Recently, advances in deep reinforcement learning have achieved successes in a variety domains such as video games, robot control and game theory. There has been an explosion of new algorithms focusing on efficiently computing stable policies. As shown in Figure 3.3, Deep Q-Network (DQN)[39] is the first breakthrough of RL which benefits from the advantages of deep learning for abstract representation in learning optimal policy by selecting actions that maximize the expected value of the cumulative sum of rewards [41]. This approach overcome the unstable issues with two strategies:

- Target Q-network.
- Replay Memory.

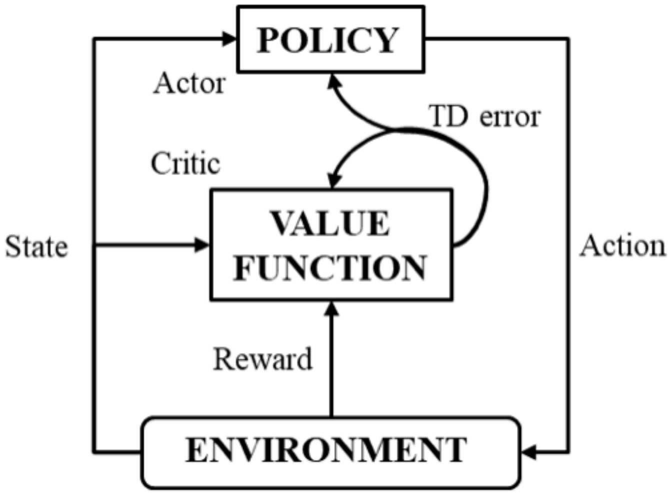

FIGURE 3.5: Actor-Critic Structure for reinforcement learning

In addition to the target Q-network and the replay memory, a serious of techniques are also introduced such as reward clipping, covolutional neural network, and RMSprop[18].

Based on the success of DQN, Double DQN, Dueling Network and Distributional DQN are proposed and the performances have surpassed human level on the benchmark of most Atari games. However, the DQN approaches are still limited by the sampling efficiency issues that makes it impractical to be applied in lager/continuous action spaces. In addition, the output of DQN is deterministic.

On the other hand, another family of reinforcement learning algorithms are developed from policy based, which parameterize policy that can select actions without using a value function[58].

The most promising approach is proximal policy optimization (PPO) [50], which performs comparably better than state-of-the-art approaches while being much simpler to implement and tune following the Actor-Critic style, as in Fig.3.3. PPO solve RL because it computes an update at each step that minimizes the cost function while ensuring the deviation from the previous policy is relatively small. This characteristic is crucial for training a human-like agents aiming to model daily travel behavior. Unlike other applications of RL, such as video games, 3D locomotion, and other forms of robots control, the time horizons of the proposed agents are relatively short, and the order of actions is also of extreme importance for output accuracy. The fundamental difference between PPO and previous Actor-Critic methods is that PPO updates its actor neural network based on the Advantage value (TD error) estimated by Critic neural network as follows:

$$A_t = -V(s_t) + r_1 + \gamma r_1 + \cdots + \gamma^{T-t} V(s_T) \tag{3.9}$$

where t specifies the time index in $[0, T]$. The actor parameters are updated by clipped surrogate objective as

$$L^{CLIP}(\theta) = \mathbf{E}_t[min(r_t\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t] \tag{3.10}$$

where $r_\theta = \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}$, and $r(\theta_{old}) = 1$. The L^{clip} decides the update rate based on the ratio of old parameter and new parameter in an interval $[1-\epsilon, 1+\epsilon]$. Such an objective assures to avoid updating too much from some noise and too slowly. A PPO algorithm that uses fixed-length trajectory segments is shown below. In each iteration, each of the N (parallel) actors collects T timesteps of data. Then we construct the surrogate loss on these N timesteps of data, and optimize it with minibatch stochastic gradient descent for K epochs [50]. Further, DeepMind also proposed a distribution computing version of PPO to improve sampling efficiency for robust policy calculation.

More details can be found elsewhere.

3.5 Trajectory Generation based on Recovered Reward Function

In the previous section, we have shown how to construct the reinforcement learning architecture and learn reward function approximation on the basis of the IRL algorithm. In this section, we present the method for generating more realistic behaviors on the basis of the constructed model and learned reward function.

First, the RL framework is based on a first-order Markov model that is not able to capture the long-term dependencies and meanwhile suffers from a memoryless problem. Meanwhile, if we treat an agent regardless of the Markovian property, the simulation result will not match the population density distribution at the citywide scale at each time step, because a memoryless agent is not able to reproduce a recurrent trip pattern such as a "home-work-home" pattern as real human beings do, and likewise fails to with other similar predictive behavior factors.

To cope with this difficulty, we draw on the idea from [39], this is, we propose a simplified history-dependent method to generate simulated trajectories that incorporate cues from the current state and time, previously visited places and behavior history produced by the function described above. The algorithm modifies standard Q-learning in the following ways to make it suitable for training a travel behavior agent.

First, we store the agent's experiences at each time-step, $e_t = (s_t, a_t, r_t, s_{t+1})$, in a data set $D_t = e_1, \cdots, e_t$, pooled over many episodes (where the end of an episode occurs when a terminal state is reached) into a replay memory [40]. In the iteration of the algorithm, we simulate multiple episodes to sample experience, $(s, a, r, s^{'}) \sim U(D)$. In practice, our algorithm only stores the last N experience tuples in the replay memory, and samples uniformly at random from D when performing updates.

Second, in formal Q-learning, an agent learns about the greedy policy $a = argmax_{a'} Q(s, a^{'})$ while following a behavior distribution that ensures adequate exploration of the state space. On the contrary, on the basis of previous research[49], human travel behavior shows an apparent recurrent pattern, indicates when people make an action decision at each time step, and shows a preference for choosing an action that has appeared in previous experiences. In other words, after several steps of actions, agents tend to exploit their previous experiences rather than exploring the new environment. To reproduce this factor, we define the behavior distribution as being selected by an $\epsilon - greedy$ policy that follows the greedy policy to explore the state space with probability $1 - \epsilon$ and exploit previous experience D_t to select an action with probability ϵ .

3.6 Summary

In this section, we formulate the reinforcement learning agent model and environment structure on citywide level, and introduce the existing algorithms for solving the problem. We notice that the MDP model for people travel behavior at citywide level would generate enormous state and action spaces that makes the problem not calculable. To solve this issue, we develop a graph-based representation of MDP that significantly decrease the state and action space. Furthermore, we discuss the application of deep reinforcement algorithm. We found that although current deep reinforcement learning algorithms have improved the data efficiency from sampling and parameter updating (i.e. Proximal Policy Optimization), the calculation time is still extremely long and the agents are usually generate meaningless episodes with unknown reason. Especially the inverse reinforcement learning (we will introduce in the next section) needs iteratively for parameter approximation.

4 Learning Human Travel Preferences from Anonymous Location Data

4.1 Introduction

The fundamental of building a successful agent based simulation is on the basis of agents are capable of reproducing realistic decision-making at human beings level. Existing agents techniques can be classified into five classes by [47] as simple reflex agents, model-based reflex agents, goal-based agents, utility-based agents and learning agents[64]. The reinforcement learning agent is a combination of utility-based agent and learning agents. As we discussed in Section 3, reinforcement learning techniques has the advantage that it allows the agents to initially operate in unknown environments and to become more sophisticated than its initial knowledge alone might allow [76]. The learning mechanism is based on the feedback of reward on how the agent is doing and determines how the behavior policy should be modified to do better in the future. However, existing reinforcement learning application are mainly focused on video game, robot control and Go, which reward and goal are fixed and known. However, in real-world application, the reward of human behavior should be regarded as unknown to be ascertained through empirical investigation [42]. The reward functions are usually consist of multiple attributes and it is hard to see how one could determine the relative weights of these terms. In transportation domain, the reward is usually be seen as utility of behaviors, the representation of the utility and the parameters can be derived from travel behavior questionnaires by attendees' active report, or maximum likelihood parameter estimation based on discrete choice model from travel behavior surveys that connected behaviors and socio-economic attributes. However, location data which is collected passively can not reveal human preferences on decision makings. Another idea of this issue is to recover the expert's reward function from their behavior trajectories and use this to generate desirable behavior. Thus approach is inverse reinforcement learning. In this section, we apply the inverse reinforcement learning problem in travel behavior decision making, and explain how to recover human travel preferences from anonymous location data.

4.2 Problem Setting

The inverse reinforcement learning (IRL) problem can be characterized informally from [46] as:

- **Given** 1) measurements of an agent's behavior over time, in a variety of circumstances, 2) if needed, measurements of the sensory inputs to that agent; 3) if available, a method of the environment.
- **Determine** the reward function being optimized.

In the inverse reinforcement learning setting, an agent is trying to sample and learn experts' behavior in target space. The agent tries to maximize mapping functions of the features for each state-action pair, $\mathbf{f}_{s_j} \in \mathbb{R}^k$, to a state-action reward value representing the feedback from environment in terms of visited state and chosen action. This function is parameterized by the weights θ. The reward value of a trajectory is simply the sum of state rewards, or, equivalently, the reward weight applied to the path feature counts $\mathbf{f}_{\zeta} = \sum_{s_j} \in \zeta \mathbf{f}_{s_j}$, which are the sum of the state-action features along the trajectories [77].

$$reward(\mathbf{f}_{\zeta}) = \theta^{T}\mathbf{f}_{\zeta} = \sum_{s_j \in \zeta} \theta^{T}\mathbf{f}_{s_j} \tag{4.1}$$

The agent demonstrates single trajectories, and has an expected empirical feature count, $\widetilde{\mathbf{f}} = \frac{1}{m}\sum_i \mathbf{f}_{\overline{\zeta_i}}$, based on many demonstrated trajectories.

4.3 Processing pipeline

RL agents interact with their environment in discrete time steps and output a series of state-action pairs as trajectories. Thus, it is necessary to extract and process data that enable observations of travel behavior decision making in the same format. In this study, the training data should satisfy the following conditions.

The training data should be represented as a sequence of time-stamped points, each of which specifies a traveler ’ s location. The training data should have enough temporal and spatial granularity to provide travelers with moving-related decision making at each time step.

In the first step it is necessary to infer travelers ’ location at each time step. Unlike most popular data sets, in this study, the GPS data collected from Yahoo Japan Corporation were used. All data were provided by users who agreed to upload their location for research purposes through a disaster alert application. The temporal granularity of this dataset is sparser than that of many other GPS datasets(such as GeoLife) which are logged in dense representation, e.g., every 1 5 seconds, but it is similar to that of various CDRs datasets used in recent mobility studies. Considering the data sparsity, most existing stay point detection methods cannot be used. The method proposed by [74] was used to detect stay points and extract travel sequences. Then, the trips ' transport modes were classified as " walk ", " vehicle " and " train " based on decision tree method following previous work [66]. After that, since we discretize the time into 30 minutes as one time step, sometimes there are multiple trips occurred in the same time step, so we choose a major transport mode for each time step using the following rules:

- if there is a trip with transport mode 'train', the majority transport mode is 'train'
- when the trips contain 'vehicle' and 'walk' modes, take 'vehicle' as majority mode
- otherwise the mode is 'walk'

Besides, to simplify the model, we set out all the single trip should be finished in 30 minutes (one time step). Therefore, the stay points and trips are regarded as states and actions (details of state and action are explained in the following section), and the demonstration trajectory of an individual i can be represented as:

$$D^i = (s_0^i, a_0^i), (s_1^i, a_1^i), \cdots, (s_T^i, a_T^i) \tag{4.2}$$

The method overcomes the data sparsity issue; thus it has the potential to be applied with many kinds of datasets that are open for research purposes.

4.4 Maximum Entropy Inverse Reinforcement Learning Method

To imitate human behavior, agents should receive a higher reward when they choose an appropriate action that human would choose in the same situation. Recovering the hidden reward from a set of demonstrations could help to understand human action preference and enable agents to reproduce higher levels of human-like actions in the simulation. Various approaches using structured maximum margin prediction, feature matching, and maximum entropy IRL have been widely used for recovering the cost function. However, recovering the agent's exact reward weights is an ill-posed problem [29]. Many reward weights, including degeneracies, for example all zeros can make demonstrated trajectories optimal.

The maximum entropy IRL approach is used as a foundation, and the model to model daily travel behavior decision-making is extended. The principle benefits of the Maximum Entropy paradigm include the ability to handle expert sub-optimality as well as stochasticity by operating on the distribution over possible trajectories. In this formulation, the probability of experts' preference parameter for any trajectories can be calculated with a probability proportional to the exponential of the reward:

$$P(\zeta_i|\theta) = \frac{1}{Z(\theta)} e^{\theta^T \mathbf{f}_{\zeta i}} = \frac{1}{Z(\theta)} e^{\sum_{s_j \in \zeta} \theta^T \mathbf{f}_{s_j}} \tag{4.3}$$

Given parameter weights, the partition function $Z(\theta)$, always converges for finite horizon problems and infinite horizons problems with discounted reward weights. For infinite horizon problems with zero-reward absorbing states, the partition function can fail to converge even when the rewards of all states are negative [77].

Second, to imitate human behavior, agents should receive a higher reward when they choose an appropriate action that another human would choose in the same situation. However, the reward r_t received from a model in terms of the transition from s_t to s_{t+1} by action a_t is hard to be estimated directly due to the dimensionality and size of the state space in real world. Instead, recovering the hidden r from a set of demonstrations could help to understand human action preference and enable agents to perform higher levels of reproducing human-like actions in the future. Various approaches using structured maximum margin prediction, feature matching, and maximum entropy IRL have been proposed for recovering the cost function [29]. We choose the maximum entropy IRL algorithm proposed by [77] and extend the model to model daily travel behavior decision-making. The principle benefits of the maximum entropy theory include the ability to handle expert sub-optimality as well as stochasticity by operating on the distribution over possible trajectories.

The distribution takes this form because the given exponential distribution maximizes the entropy subject to a fixed mean value. In this study, we use the following linear parameterized representation: $R(s,a) = f(s,a)^T\theta$ Then we can apply f to every state-action pair in the demonstrations. Every ζ_i in demonstration D is a sequence of T-step state- action pairs, the feature space $\phi : \mathbb{R}^N$ can then be applied to the trajectory as $R(\zeta|\theta) = \sum_{(s,a)\in\zeta} \phi(f(s,a)) = \phi^T f_\zeta$

In this study, the basic hypothesis is that agents visited the states with similar features should receive the similar rewards from environments. In order to reach this objective, we use the data provided by National Land Numerical

Information to approximate the reward function. The National Land Numerical Information is a series of open data that in terms of information related to national lands to support the promotion and formulation of land planning such as the Comprehensive National Development Plan, National Land Use Planning, and National Spatial Strategy[3]. We consider multi-dimensions of characteristics: residential population at night, number of offices and schools, road density, employee population to characterize the destination location. Travel distance and travel time are used to represent the cost features for actions with respect to different transport modes.

The problem of deriving an optimal reward weight vector ϕ from demonstrated trajectories based on IRL can be formulated by maximizing the joint posterior distribution of observing expert demonstrations $\mathcal{D}$ under a given reward structure and of the model parameters θ.

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\subseteq} = \log P(\mathcal{D}, \theta | r) + \log P(\theta) \tag{4.4}$$

This joint log likelihood is differentiable in terms of the parameters θ of the linear reward function, and the can be optimized by gradient descent methods. As be presented in [77], the gradient of the expert demonstration term $\mathcal{L}_{\mathcal{D}}$ in terms of the reward function parameters is equal to the difference in feature counts along the trajectories which can be represented as:

$$\mathcal{L}_{\mathcal{D}} = \tilde{f} - \sum_{\zeta} P(\zeta_i \mid \theta) f_{\zeta_i} = \widetilde{f} - \sum_{(s,a)\in\zeta_i} G_{(s,a)} f(s,a)^T \theta \tag{4.5}$$

where $\tilde{f}$ is the expected empirical feature counts and $G_{(s,a)}$ is the expected state-action pair visitation counts for learned possible trajectory distribution.

The detail of algorithm is summarized in Algorithm 2, with the loss and gradient given by the linear Maximum Entropy formulation. The expert's state action frequencies μ_D^a, are summed over the actions to compute the expert state frequencies $\mu_D = \sum_{a=0}^{A} \mu_D^a$.

We found that few state-of-the-art IRL studies have taken time factors or time series into consideration because standard benchmarks and previous tasks such as urban navigation and activity forecasting are not sensitive to time change. However, we found that travel behavior is highly correlated with time. For instance, business areas are more attractive in the daytime for commuters than in the nighttime. Likewise, working schedules for most people result in peak transportation in the mornings and evenings during weekdays. Despite such typical correlations between time and travel behavior, certain issues will cause unrealistic travel behavior. Unfortunately, the linear function makes it difficult to model the influence of time change because all features are highly correlated to time change. To tackle this problem, we proposed a time-dependent IRL algorithm, that can separately train the reward function weights for a discretized time MDP model at each time step. Furthermore, unlike common IRL problems that utilize infinite horizon MDP on the basis of a standard value iteration algorithm, we introduce finite-horizon MDP to better represent human daily travel behavior and describe the solution in Algorithm 1. Finally, the expected action visitation frequencies can be computed by enumerating all paths and counting the number of paths and times in each path in which the particular state-action is chosen [56].

Algorithm 2: Expected state action visitation counts calculation

```
Input  : The feature parameter vector $f$, state space $S$, action space $A$,
         initial state $s_0$ and terminal state $s_g$
Output: Expected state action visitation frequencies $G_{s,a}$

Backward Pass;
Set $Z_{s_i} = 1$ for valid goal states, otherwise 0;
Recursively compute for $T$ iterations;
$Z_{a_i} = e^{-f(s,a)} Z_{s:a_i}$;
$Z_{s_i} = \sum_{a_j \in s_i} Z_{a_j} + 1$;
Forward Pass;
Set $Z'_{s_i} = 1$ for valid goal states, otherwise 0;
Recursively compute for $T$ iterations;
$Z'_{a_i} = e^{-f(s,a)} Z'_{s:a_i}$;
$Z'_{s_i} = \sum_{a_j \in s_i} Z'_{a_j} + 1$;
Summing frequencies;
$G_{s,a} = \frac{Z'_{s_i} e^{-f(s,a)} Z_{s_i}}{Z_{s_0}}$
```

Algorithm 3: Feature expectation calculation

```
Input  : initial state $s_0$, state space $S$, stochastic policy $\pi(a|s)$
Output: Expected feature counts, $E[\mathbf{f}]$ under the stochastic policy $\pi(a|s)$
Compute $D_{(s_x,a_y)}$ from initial state under Algorithm 2

$E[\mathbf{f}] \leftarrow 0$;
for $s_x$ in $S$ do
    for $a_y$ from $s_x$ do
        $E[\mathbf{f}] \leftarrow E[\mathbf{f}] + D_{s_x,ay} \pi(a_y|s_x) \mathbf{f}_{s_x,a_y}$
```

```
Algorithm 4: Maximum Entropy Inverse Reinforcement Learning
Input  : The demonstration state action frequencies $\mu_D^a$ , feature parameter
         vector $f$, state space $S$, action space $A$, and discount factor $\gamma$
Output: Optimal reward function parameter $\theta^*$
Initialize $\theta^0$ with random number;
Iterative model refinement;
for $n = 1 : T$ do
    $r^t = f(s,a)^T \times \theta^t$;
    Solve MDP with current reward;
    $\pi^t = \texttt{approximatevalueiteration}(r_t, S, A, \gamma)$ ;
    $G_{s,a} = UseAlgorithm2(\pi^t, S, A, T)$;
    Determine Maximum Entropy loss and gradients;
    $\bigtriangledown \mathcal{L_D}^t = (\mu_D^a - G_{s,a}) \times f(s,a)$;
    Update parameter with gradient;
    $\theta^{t+1}$ = update weights with $\theta^t$, $\bigtriangledown \mathcal{L_D}^t$
```

4.5 Feature Engineering of Reward Function

Incorporating feature construction into IRL has been recognized as an important problem for some cases. It is often easier to enumerate all potentially relevant component features than to manually specify a set of features that is both complete and fully relevant. For example, when emulating a human driver, it is easier to list all known aspects of the environment than to construct a complete and fully relevant reward basis.

In this study, the reward structure is restricted by stipulating that states with similar features should have similar rewards. From this perspective, the data provided by National Land Numerical Information were used as features to approximate the reward function as mentioned before. Multiple-dimensions of characteristics were considered: night population, number of shops, road density, number of employees, and land use to characterize the state. Travel distance and travel time were used to represent the cost features for actions with respect to different transport modes.

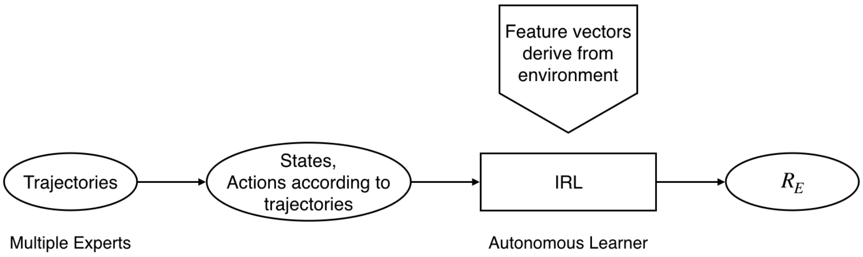
Feature vectors derive from environment
Trajectories
States, Actions according to trajectories
IRL
R_E
Multiple Experts
Autonomous Learner

FIGURE 4.1: Work flow of reward function estimation via inverse reinforcement

4.6 Estimation Results

In this section, we show the estimation results of reward function based on inverse reinforcement learning. The experiments are implemented in Tokyo metropolitan area, and the People Flow data which conducted from Person Trip Survey in Tokyo are used as training data. This dataset contains about 700,000 peoples' daily trips in a typical work day in 2008, and all the trips are assigned to transportation network and interpolated in 1 minute. The process flow is shown in Fig. , we randomly choose multiple people's trajectories as experts data for each experiments.

The first problem we need to figure out is how many trajectories are needed for inverse reinforcement learning algorithm to derive stable reward function parameters. We set 10 experiments with different training data amount from 100 to 1,000. As shown in Fig. 4.2, we visualized the reward function parameters over time with different training data amount. It is obviously that

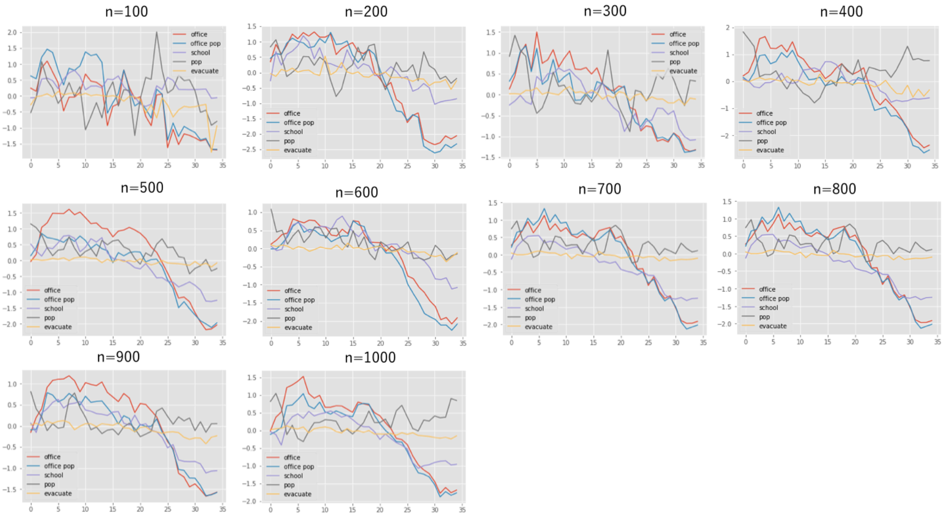

FIGURE 4.2: Estimation results with different samples

when training data is less than 500, the parameter curves show erratic fluctuation that is not accordance with real people's behavior preferences. Take the experiment result (n = 100) as example, the night population density feature curve fluctuates dramatically in a short time period, which means the preference of choosing a place with a lot of houses, changes at each time step. As the training data amount increase, the parameter curves become smooth and reasonable. As shown in Fig. 4.2(n=1000), the weight of office count feature increase in the morning peak period and become to decrease from afternoon, on the contrary, the weight of night population shows totally opposite trend. This result reflects a normal commuter's daily travel pattern (which is also easily well observed from the dataset). Thus, in the following part of this thesis, we choose samples=1000 to infer reward function.

Furthermore, we test the differences of reward function parameter between

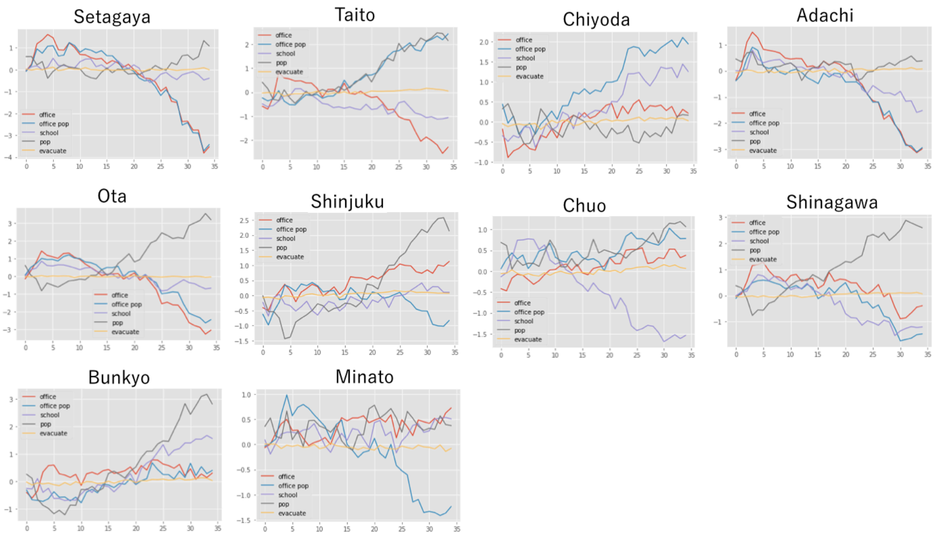

FIGURE 4.3: Estimation results with samples derived from different area

different area. Using the same dataset, we classify trajectories based on their initial state (location at 0:00, regard as surveyee's home) by 23 zones of Tokyo, and for zone we randomly choose 1000 trajectories to train the reward model.

As shown in Fig. 4.3, we can easily see that the reward function curve shows totally different pattern between different zones. 'Setagaya' locates at west of Tokyo and has the highest residential population density, the curve of office count(residential population density) feature's weight increases(decreases) in the morning and decreases(increases) after morning peak. The results reflect a typical commuter daily movement pattern, that people from this area could receive higher reward by choosing commercial areas (where have more office facilities and less residential population) in the morning, and show opposite pattern in the evening. Another pattern can be seen in Fig. 4.3 ’Shinjuku’, where is the major commercial and administrative centre, housing the northern half of the busiest railway station in Tokyo. The reward function curves show opposite pattern comparing to the result of ’Setagaya’. In other area such as ’Bunkyo’ and ’Chiyoda’, the weight of ’school’ feature becomes more influential than other areas. From this perspective,

Previous studies have shown that human mobility could change in rare events such as disaster and big events[16]. To clarify the differences of reward function between different scenarios, for each area (Tokyo and Hiroshima) we choose a rare scenario to compare with normal scenario to see the differences. For the Tokyo area, we choose the day on January,22,2018 when a heavy snow started to fall at around 15:00 and caused severe traffic congestion. We can easily observe the changes of reward function curves happened after 16:00, the weight of travel time cost decreased dramatically which shows people done their utmost to avoid long time trips, and the weight of evacuation places turn to be higher value than usual.

Another example is the heavy rain happened in Hiroshima, July, 7th, 2018. As the heavy rain have continued for a couple of days, the reward function curves show a totally different pattern from morning. The residential population density becomes the major influential factor, means that more people would choose to stay at home.

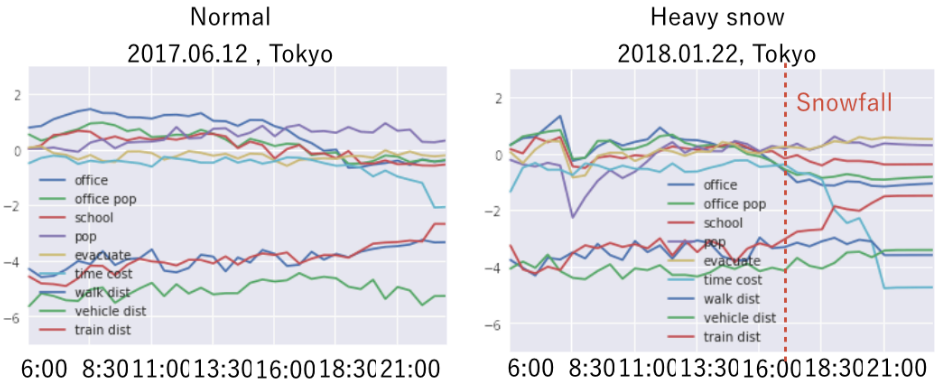

FIGURE 4.4: Comparison of estimation results from a normal day (left) and a day with heavy snow (right) in Tokyo

4.7 Summary

In this section, we explain the workflow of inferring reward function from anonymous location data. We define the form of training data that inverse reinforcement learning algorithm could use and present the pipeline of data processing to solve the data sparsity issue. Then, we discuss the relationship between the training data amount and estimation results. We also recover the reward function from different area in Tokyo Metropolitan area and different scenarios such as normal day and rare events(heavy snow and rain). The different patterns of recovered reward functions

There are still few tasks remaining in inferring reward function. First, in this study, we use simple linear model to represent reward function, although it successfully revealed the relationship of multiple features and final reward,

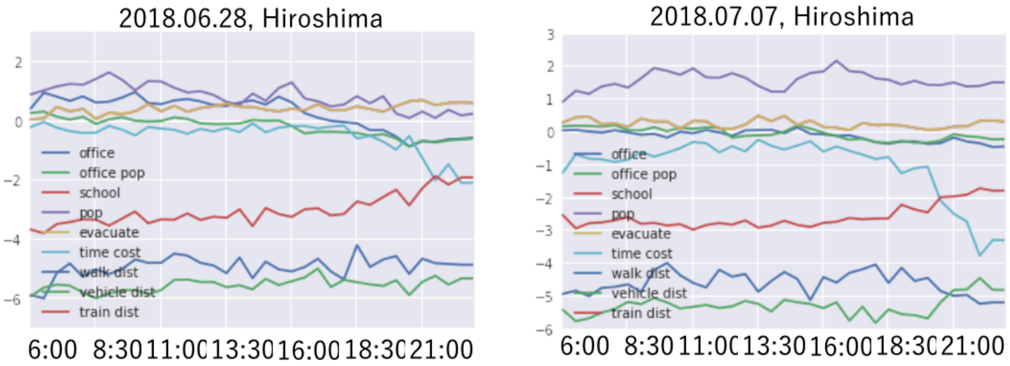

FIGURE 4.5: Comparison of estimation results from a normal day (left) and a day with heavy rain (right) in Hiroshima

some features such as week of the day, season, weather are hard to be added to the current model, such limitation constrains us to infer long-term (1 week or 1 month) reward function, or generalize the current reward function to different scenario. Second, feature engineering is another direction to improve the estimation results. During the training, we found several features such as land use, public facilities (hospital, post office, welfare) have little influence on reward function, some features such as evacuation shelters, schools only have effect on specific scenarios, further studies on selecting effective features are needed. Third, our current inverse reinforcement learning methods randomly samples demonstration trajectories from data set to represent a typical agent model which reveals common activity pattern. Although training multiple sets of parameters could reveal measurable population heterogeneity, how to represent real population structure with agent models is still a open problem.

5 Application of Reinforcement Learning for People Flow Simulation

5.1 Introduction

In this section, we apply the reinforcement learning agent model developed from Section 3 and reward function that recovered from Section 4 using the data collected from GPS data and People Flow Data for real world people flow simulation in Japan. We first introduce the simulation framework. Then, to better evaluate the proposed modeling framework, we simulate daily people movement based on the developed model and compared the simulation results with ground truth. Furthermore, to test the agent performance on different situations, we trained the agent models using demonstration data that collected from a normal day and a disaster day separately, and examine the simulation results. Another important application of the agent-based modeling and simulation framework is to forecast people movement in unprecedented scenario. Lack of historical data is the major challenge for direct forecasting because the collection and storage of emerging data sets are just started from recent years. We focus on the case that the same event (i.e. disasters) have happened at other places, where the data of people movement is sensed and stored. We use the mobility data collected from other places and develop the agent models, then simulate people movement in target area.

5.2 Experiment and Evaluation

5.2.1 Simulation Settings

In this study, the agent models do not refer to any specific people, it is difficult to evaluate the generate trajectories or single action choice because we cannot find and pair a ground truth trajectory from any dataset for comparison. Instead, we evaluate the mass people movement simulation based on population distribution and passenger amount comparing to the aggregated ground truth. Fig. 5.1 shows the overall simulation and evaluation workflow of this study. We first split the available data set into two part as ground truth and training data. The training data is used for estimating real people's reward function following the algorithm explained in Section 3. The volume of training data volume is up to 20 percent of the dataset. On the other hand, we derive the population distribution at the beginning of the day as simulation initial distribution. Each agent is assigned with a reward function that motivated its actions. Simulation starts at 6:00 in the morning and ends at 23:30 when last action choice is decide. Time step is set as 30 minutes. At last, the simulation results are compared with the ground truth data which share the same initial state and amount.

5.2.2 Baselines and matrices

The method was compared with the following baseline models.

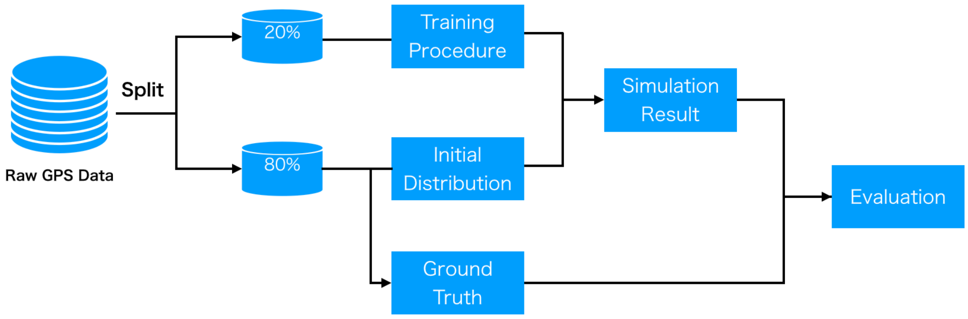

FIGURE 5.1: Simulation framework for evaluation

- First-order Markov chain[55]: this model defines the current state as the current location, and the next step choice probability is only dependent on current location with space-complexity where is the number of the total locations
- Time-dependent Markov chain Model[52]: it assumes the transition probability is time-dependent by splitting time into multiple time periods (e.g., morning, afternoon, evening). This setting was further extended, and a specific timestamp was added as a time feature to improve performance.
- Discrete Choice Model[13]. DCMs are a de facto standard in practical evaluation of the travelers ' population response to system parameters and policy interventions. In practice, planners operate with parametric DCMs to learn traveler 's choice preferences and determine how travelers trade of various attributes of a given set of travel choice alternatives.

- Neural network actor:
- Recurrent neural network actor（[73]）: We also implemented the state-of-the-art recurrent neural network as actor for temporal prediction, which has been successfully applied in word embedding and ad click prediction.

To evaluate the performance of proposed model, negative log-loss (NLL) was used as the probabilistic comparison metric. The NLL:

$$NLL(\zeta) = E_{\pi(a|s)}[-log\prod_{t}\pi(a_t|s_t)] \tag{5.1}$$

is the expectation of the log-likelihood of a trajectory ζ under a policy π. In the example, this metric measures the probability of drawing the demonstrated trajectory from the learned distribution over all possible trajectories. The distance between two trajectories was also calculated as a physical measure the distance error. Given two trajectories A and B with the same number of points, $Distance(traj_A, traj_B) = avg(dist(p_i^t, p_j^t))$ is the Euclidean distance between point a and point b, and n represents the uniform discrete time slot. Finally, the Jaccard similarity coefficient was used as accuracy

5.2.3 Datasets

Yahoo Japan Corporation collects the GPS data of each individual who has agreed to provide their location data for research purposes through the disaster alert application. Each GPS log consists of the ID number, timestamp, longitude, and latitude. The dataset is started from 2014 with around 1 million users (an approximate sample rate of 1% from all over Japan) . Both Android and iPhone users' data are collected by the standard modules that are commonly used for location data collection. For Android phones, they automatically update location in every 30 minutes information when the service function is active, and each point is continuously observed until the location converges to raise its accuracy of the location. The mechanism of data collection of iPhone is different, user ’ s location is detected when the phone stops to stay at current location and starts moving. Unfortunately, this dataset does not have a “ golden standard (e.g. taxi cab trajectory data observed every 10 seconds with less than 10 meters error) ” where the data tells us the exact trajectories of how the users are moving. However, studies have shown that collectively analyzing such dataset could provide valuable insights on urban dynamics, and also on individual behavior. The temporal granularity is more closed to that of various call detail records (CDR) datasets , but is sparser than many taxi trajectory datasets [67]. Our method could be applied with various GPS or CDR data from various parts of the world because it only needs to observe the beginning and end of the users ’ commuting movement, which could be achieved using various datasets other than our GPS dataset.

5.2.4 Experiment on Normal Scenario

Because the mobility of each individual is unique in the geographical space, to examine model performance in different urban layout, the model instantiated in two different areas in Japan as shown in Table 1. Tokyo comprises Japan ’s largest domestic and international hub for rail, ground, and air transportation. The transport network in the Tokyo area includes public and private rail and highway networks; airports for international, domestic, and general aviation; buses; motorcycle delivery services; pedestrians; bicycling; and commercial shipping. Commuter rail ridership is very dense, at 6 million people per line mile annually with the highest utilization among automotive urban areas. To verify that this approach is applicable to different types of urban situation, another case study was set up in the Hiroshima eastern area. The total population and urban density are less than Tokyo area. Transport in eastern Hiroshima is also different from that in Tokyo, because only one train line connects this area with central Hiroshima.

From the Yahoo GPS dataset, users who provide sufficient data points were first extracted. The rate of time slots (30 min as unit) a user was observed (as GPS logs have reported) out of the total slots number in a day was set. It was found there is significant signal loss during the period between 0:00 to 5:00 because of the phone being turned off or/and people staying inside high buildings for a long time. Thus, the simulation time period was set between 6:00 to 23:30 for better model representation and performance.

The observed population was approximately 100,000 in Tokyo and 5,200 in the Hiroshima eastern area from a normal observed weekday. In addition, 1,000 trajectories were randomly chosen for training, and use remaining trajectories (more than 80 percent) as the test set. To evaluate the accuracy, the synthetic data followed the same population distribution (initial location distribution) and population size as the test set and were compared with some metrics explained in the next subsection.

Table 5.1 shows how the proposed method outperformed at a level that was comparable to that of other baseline models. Markov models have shown very high performance in this task compared with other methods. Scores in Hiroshima were better than in the Tokyo area. This does not mean that training in Hiroshima was more successful but the area in Hiroshima is much smaller than in Tokyo, agents are faced fewer choices when choosing actions, and synthetic trajectories were much closer to real trajectories.

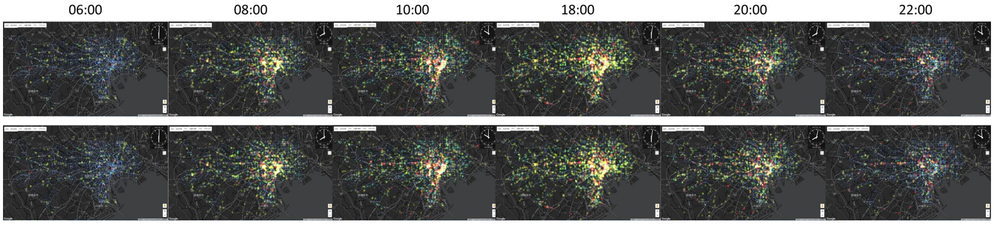

FIGURE 5.2: Visualization of simulation results in Tokyo area on normal day (upper: simulation, bottom: ground truth). Each dot presents an agent, and the color of dot represents agent's transport mode. Blue: stay, Yellow:walk, Green: vehicle, Red: train

TABLE 5.1: Performance evaluation on individual trajectory generation

	NLL		Distance Error(km)		Accuracy	
	Tokyo	Hiroshima	Tokyo	Hiroshima	Tokyo	Hiroshima
First Order MC Model	12.43	3.20	4.54	3.28	0.35	0.37
Time-dependent MC Model	10.45	2.91	3.67	2.50	0.39	0.40
Neural network actor	-	-	-	-	0.025	0.023
Recurrent network actor	-	-	-	-	0.12	0.14
Proposed Method	8.72	3.56	3.08	2.08	0.43	0.52

We first show the visualization results of trips generated by proposed framework and ground truth if Figure 5.2. Agents are presented by dots with different colors to distinguish transport mode. Comparing with the ground truth, it is obvious that the simulation results are highly similar to a real traffic situation. Besides, the agents population distribution is coincide with the ground truth during daytime. We also observed that an inner circle was generated by the railway trajectory in the center of the city and that a clear connection was formed between east and west Tokyo.

The main goal of this research was to enable RL agent to generate synthetic trajectories without compromising the observation data (i.e. raw GPS data or products from them). However, it is an extremely hard task to infer whether a synthetic trajectory is “ accurate ” or “ realistic ” because it is not known which specific trajectory in the test data should be compared. Based on synthetic trajectories, one can easily calculate the population distribution over time to examine whether agents are locating in correct locations compared with real data. The population distribution is also an important source for travel demand estimation and human mobility management. In Fig.5.3, the population distribution tested for two study area from 6:00 a.m. to 11:00 p.m. between the test dataset (x-axis) and synthetic dataset (y-axis) is shown. The correlation coefficients were higher than 0.8 over all time periods in Tokyo.

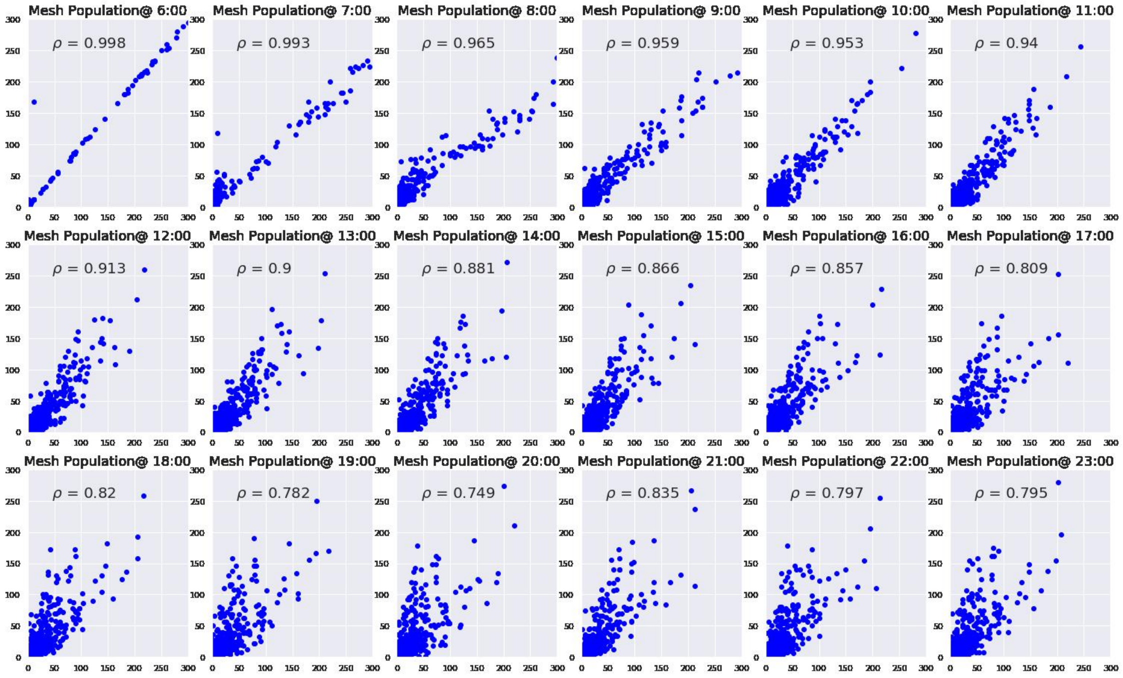

FIGURE 5.3: Scatter plots of Population Distribution in Tokyo

Also in the Hiroshima area as shown in Fig. 5.4, the distribution of the mesh population even had a better performance during commuting hours.

The population distribution with Root Mean Square Error (RMSE) and Root Mean Square Percentage Error (RMSPE) is shown in Fig.5.5 and 5.6. It was found that the errors increased significantly in commuting hours in the Tokyo area where the commuting behaviors are much more complex and frequent than in the Hiroshima area at the same time.

Another output of the simulation result that is a concern is the transport system usage situation. An accurate prediction of public transit and road network usage can play a vital role in transportation management. Different transport modes users were calculated based on a mesh unit to examine whether agents are choosing correct actions over a period of time. Fig. 5.6 and 5.7 show the vehicle users and train users in the Tokyo area compared

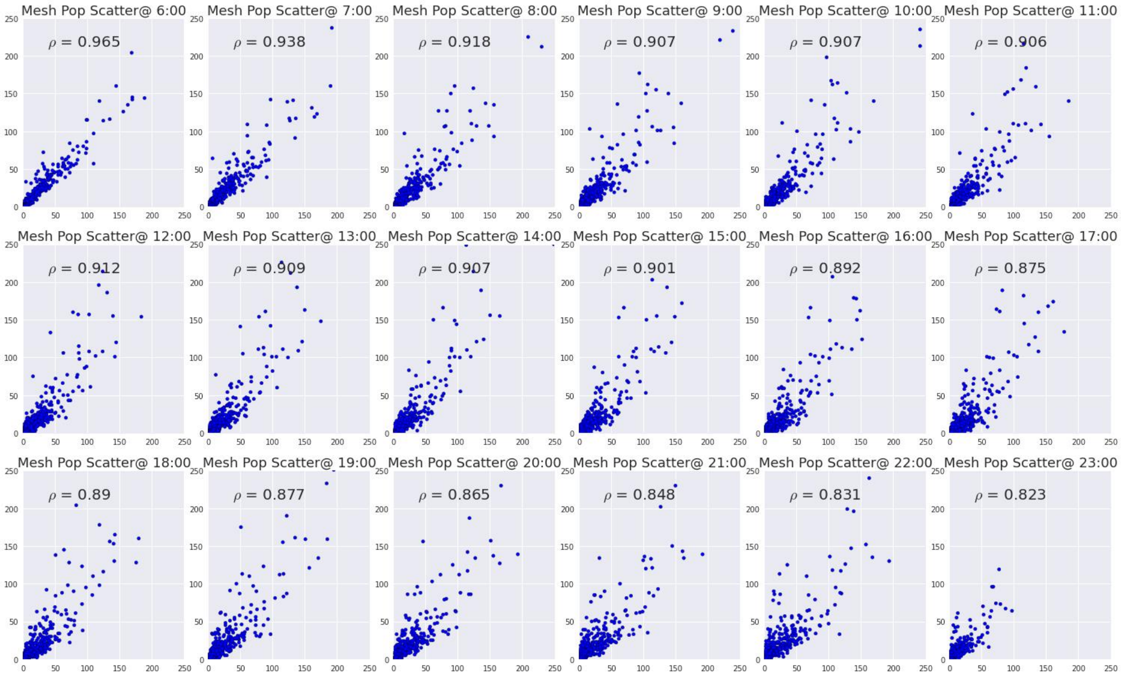

FIGURE 5.4: Scatter plots of Population Distribution in Hiroshima on Normal Day

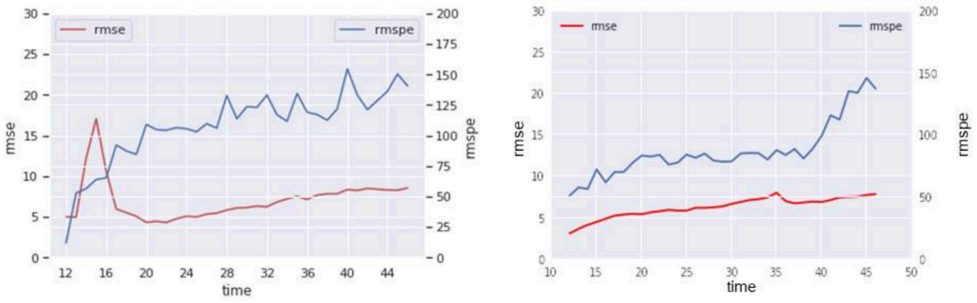

FIGURE 5.5: Comparison of mesh population accuracy by RMSE

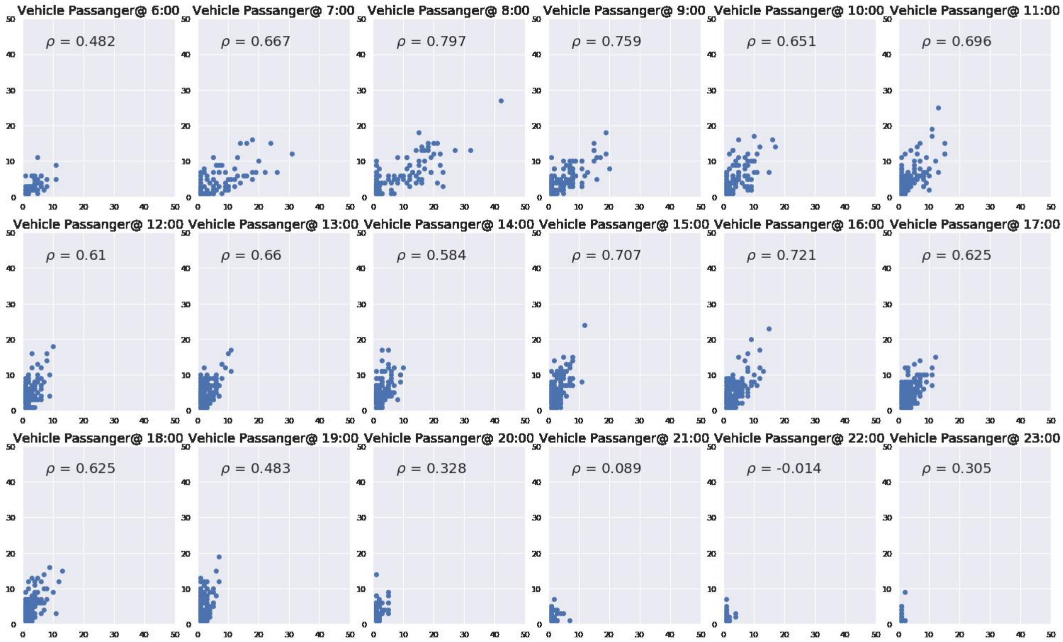

FIGURE 5.6: Scatter plots of vehicle users in Tokyo on Normal Day

with a test dataset. The results show strong positive correlation with the test dataset in commute hours. Because of the total number of users ' , one can clearly see that the railway plays a primary role in urban transport in the Tokyo area. It was also noticed that the correlation decreased dramatically at the end of the day (from 9:00 p.m.). The movements were overestimated in this period because some agents struggle with finding the way home. This result also corresponds to the dispersed pattern of population distribution at the same time.

However, the transport usage in the Hiroshima areas shows a totally different pattern. There is only one railway line in the study area and railway users are hardly observed in the training/test dataset. The vehicle users and total passengers (by train, vehicle and walking) results are shown in Fig. 5.8 and Fig. 5.9.

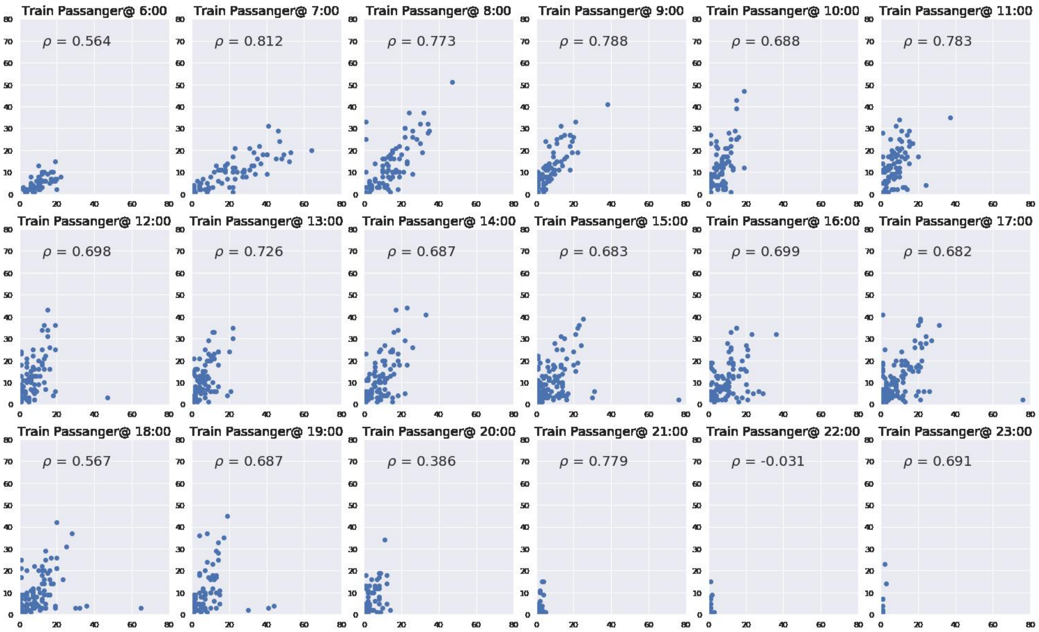

FIGURE 5.7: Scatter plots of railway users in Tokyo on Normal Day

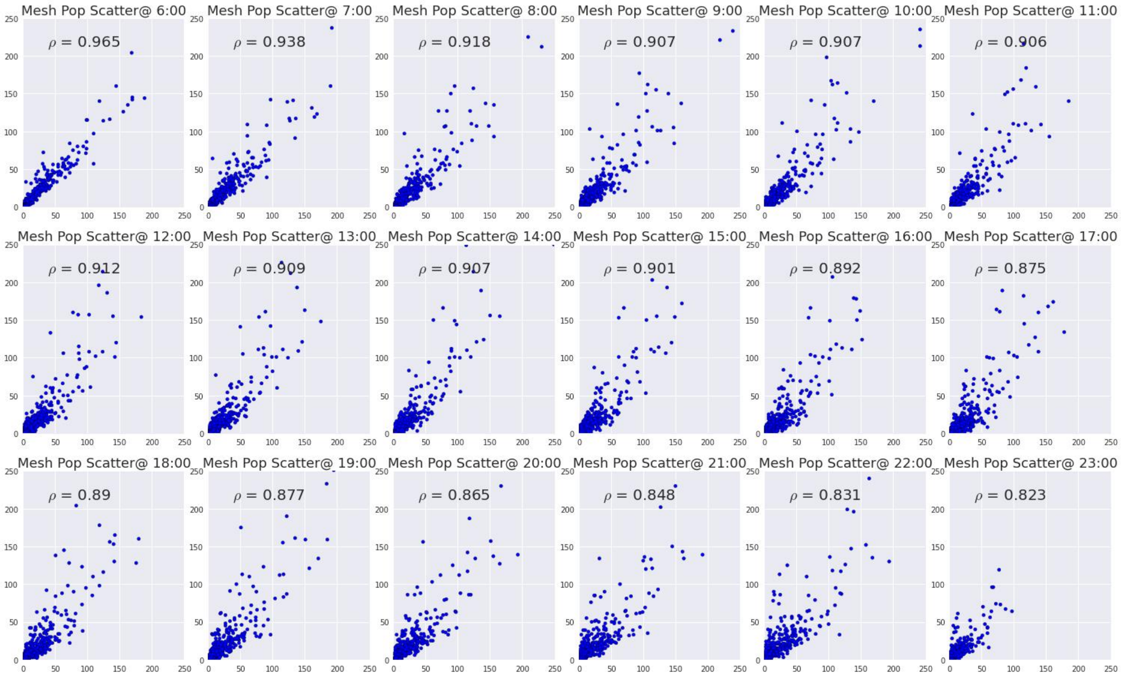

FIGURE 5.8: Comparison of accuracy of vehicle users by RMSE and correlation coefficient using vehicle (left) and train (right)

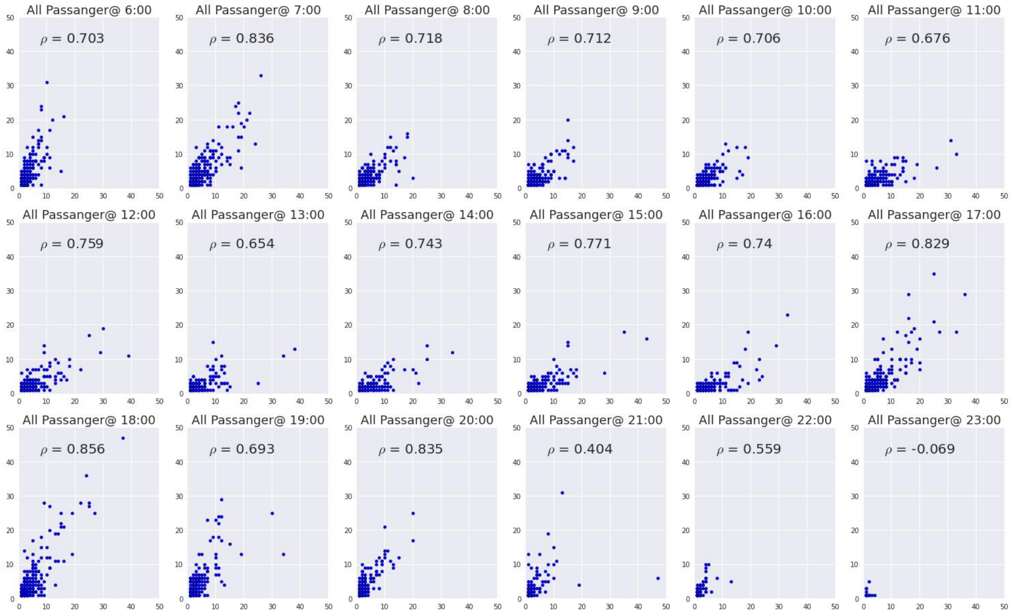

FIGURE 5.9: Comparison of accuracy of all passengers by RMSE and correlation coefficient using vehicle (left) and train (right)

In this research, IRL was used to connect ABM and anonymous locational data, and the agents ’ behavioral rules depend highly on the demonstration trajectories used for recovering reward function. Because personal attributes are unknown, trajectories are randomly chosen from the dataset, and the performance of using different numbers of demonstrations for agent model training was tested.

5.2.5 Experiment on Rare Scenario

To further check the agent model performance and the potential of applying models to some rare scenarios, in this section, we set the experiment of reconstructing people flow on scenarios. Specifically, we choose the successive heavy downpours happened in southwestern Japan in late June through

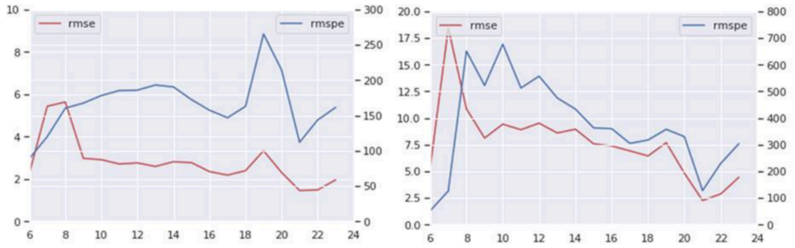

FIGURE 5.10: Comparison of accuracy of vehicle users by RMSE and correlation coefficient using vehicle (left) and train (right)

mid-July 2018, which resulted in widespread, devastating floods and mudflows. As of 20 July, 225 people were confirmed dead across 15 prefectures with a further 13 people reported missing. More than 8 million people were advised or urged to evacuate across 23 prefectures. It is the deadliest freshwater flood-related disaster in the country since the 1982 Nagasaki flood when 299 people died.

Obviously, people movement are badly effected by this event. As shown in Fig. 5.11(left), the probability distribution of trips on Normal day and Disaster day are totally different. Thus, we choose the data collected from July 6, 2018 to train the agent model, and launch people flow simulation to examine whether agent models are capable of reconstructing the phenomenon. Unlike environments of normal day scenario, the weather conditions are dramatically changing during the disaster. Here, we introduce weather data including temperature, rainfall volume, moisture and sunshine as extra dynamic features to represent urban environment.

Based on the analysis results above, to achieve higher simulation performance on rare scenarios such as heavy rain days, the agents used for people flow simulation should learn from similar scenarios, not from normal days like experiments one. So in this experiment, we try to train agent model from historical data (trajectories collected from a previous day that similar events happened). We found On 20 August 2014, Hiroshima was also struck by a series of landslides following heavy rain, which is similar with what happened in 2018. So we choose 1,000 people's daily movement from 20 August 2014, and recovered reward function on that day to learn people travel behavior on a heavy rain day. Then, we randomly choose 5000 people's trajectories from 6 July 2018 as target and run the simulation.

In Fig.5.12, the population distribution of the disaster day on study area from

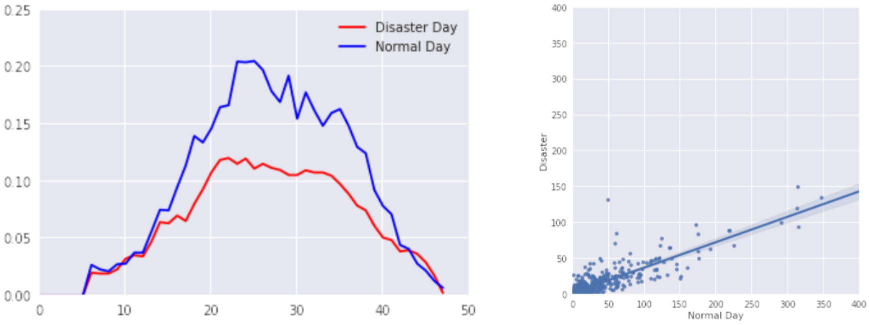

FIGURE 5.11: Mobility Differences between a Normal Day on 2018.06.28 and a Disaster Day on 2018.07.07

6:00 a.m. to 11:00 p.m. between the test dataset (x-axis) and synthetic dataset (y-axis) is shown. We found that the correlation of population distribution decreased heavily from 8:00 p.m. and end with 0.734 which is lower than the first experiment. There may be three reasons for this problem. First, the rainfall distribution is different between the training data and ground truth data. 20 August 2014, the rain started at around 19:00 and continued to the next day, so people who have already returned to home may not be influenced by the rain, and the evacuation was reported from the next day morning. On the contrary, the rainfall on 6 July 2018 starts from morning and went to peak at 7:00 p.m.. Our reward function cannot take rainfall as features into consideration right now, and that affect the results accuracy. Second, in population one, we set a clear goal that agents should return to their home at the end of the day, however, in this experiment, considering the potential evacuation and abnormal behaviors, we did not set this strong control rules for the agents. Third, there is a four years time gap between training data and ground truth data, during this period,

We further checked the places where population estimation is not corrected from ground truth. As shown in Fig 5.13, the over estimated places are mainly located in central area in Hiroshima, where the more infrastructures and high road density make the places more attractive for agents even in late night. We found the errors are mainly caused by the stay out of home issues, that the agents come from Eastern Hiroshima do not go back to home at night. Comparing to normal situation, the ratio of such kinds of agents are higher. One possible reason maybe the weather features (i.e. rainfall, temperature and so on) introduced in this experiments affect the agent's choice.

We also calculated the RMSE and RMSPE of esimation results of vehicle passengers and all transport mode passengers amount in Fig. 5.14. The x-axis

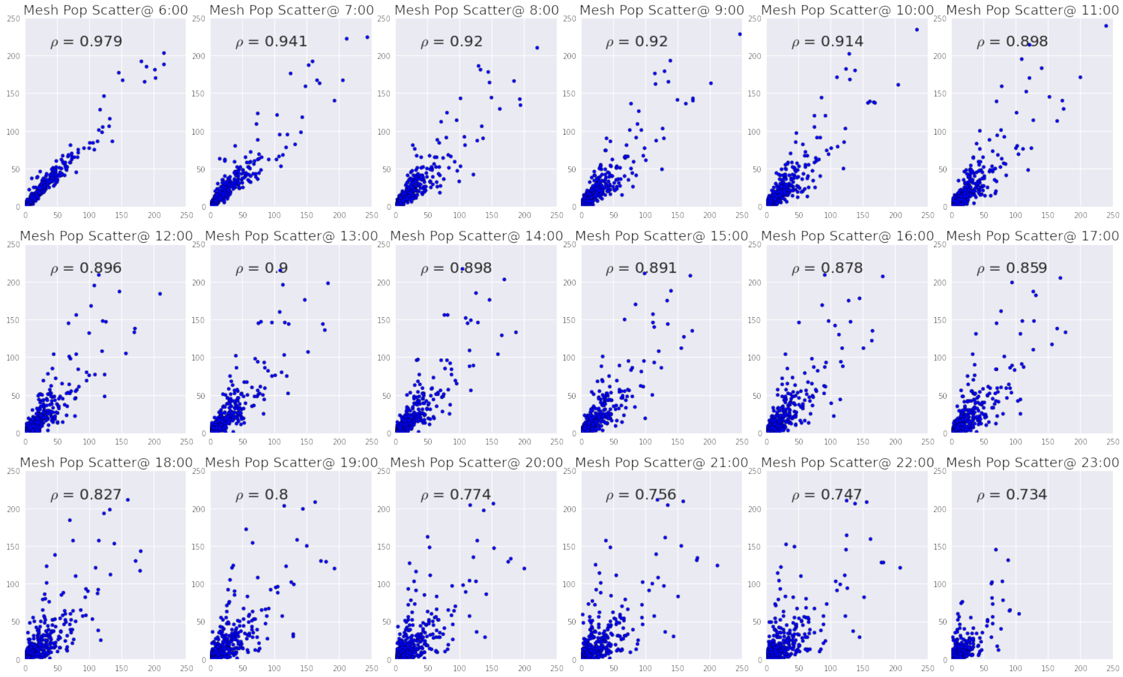

FIGURE 5.12: Scatter plots of Population Distribution in Hiroshima during Heavy Rain

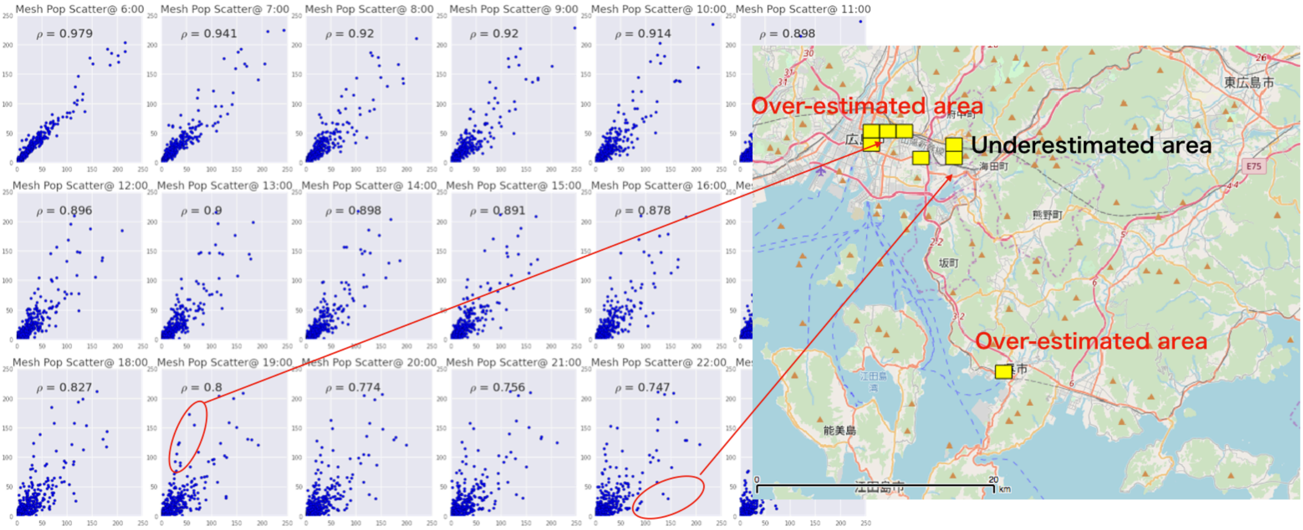

FIGURE 5.13: Over-estimated places from simulation result in Hiroshima area during Heavy rain

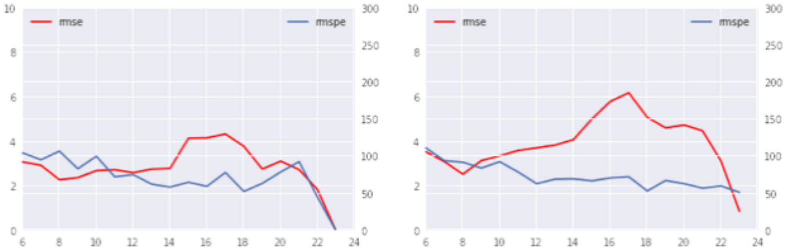

FIGURE 5.14: RMSE and RMSPE of vehicle passenger estimation result(left) and all transport mode passenger estimation result(right)

represents time stamp, left y-axis represents the RMSE population and right y-axis represents the percentage of RMSPE. It shows that the most fatal error appears during the time period between 14:00 and 19:00, when target scenario suffered heavy rain but training scenario doesn't.

In summary, rare events people flow can be simulated based on proposed method using the data collected from the past and achieved considerable accuracy. However, in this experiment, the training and target area are set as in the same place, to validate whether agent model could behavior successfully in different environment.

5.2.6 Transfer pre-trained knowledge to new area

In experiment 2, we state that training data and simulation should be in the same area. However, similar rare events could barely happened in the same area, so generally when we want to know what will happen if an event or disaster occurred in our city, we usually find where the similar event happened and learn the lessons from that city. Especially the large amount data collection technologies are just emerging in the past few years, which makes that even more difficult to collect enough data for developing and estimating agent model. From this point of view, we also want to know whether proposed reinforcement agent model can be generalized and applied into new areas.

In reinforcement learning domain, seldom studies have examined the generalization or reusable of knowledge for learned policies. Although some practitioners have started to leverage some machine learning techniques such as dropout and regularization to improve the generalization capabilities on benchmarks, there is still no real world applications on this topic. Furthermore, as discussed in section 3, the deep reinforcement learning is still immature for real world application, the current methods for transfer learned policies from reinforcement learning is impracticable.

In this section, instead of transfer learned polices, which is the result of reinforcement learning, we innovatively propose to transfer reward function from training area to target area. We suppose that the representation of reward does not change between different locations, and reuse the trained parameters in target area to learn new polices.

To test the ability of transferring learned knowledge to new area to reproduce robust prediction of people flow, we present the evaluation pipeline as

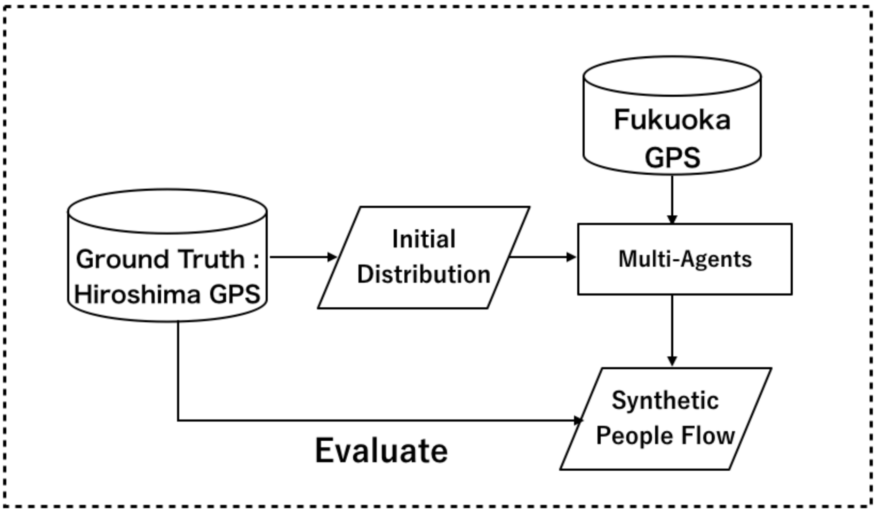

FIGURE 5.15: Evaluation pipeline for testing agent model transferable ability

shown in Fig 5.15. The experiment is under the assumption that we cannot derive GPS data from Hiroshima area for the event like heavy rain (This is very common since the history of large scale GPS data is still very short, and lot of applications are facing the same time), so the agent for 'disaster scenario' cannot be trained. On the contrary, there has happened a similar event (severe typhoon) in Fukuoka one years ago, and people movement are observed by GPS data. So in this experiment, we try to leverage the agent model learned from Hiroshima to simulate Fukuoka's people flow when both of the two places suffered severe rainfalls.

We first show the scatter plots and correlation coefficient between simulation result and ground truth in Fig. 5.16. Comparing to the simulation result that conduct from 'previous Hiroshima data' model, we found the daily estimation results decreased a little but the night estimation accuracy was improved. This result can be explained in twofolds. First, as we show in Section 4 Fig. 4.3, people from different places in a city may show different behavior preferences that we presented in the form of reward function. The differences between Hiroshima and Fukuoka may even bigger that the zonal difference

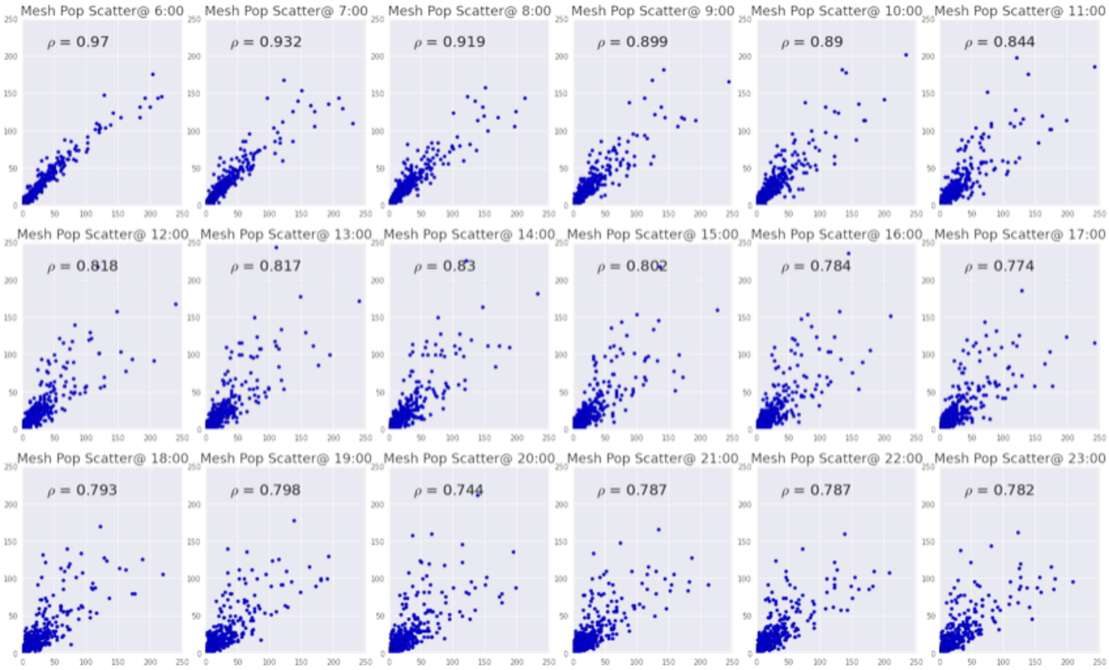

FIGURE 5.16: Scatterplots of simulation results and ground truth data in Hiroshima motivated by the reward function learning from Fukuoka

that presented in Tokyo. Driving by 'Fukuoka' reward function, agents in Hiroshima may choose different destinations comparing with ground truth of Hiroshima. Second, comparing to the training data that used in experiment 2, the training data from Fukuoka is from a similar rainfall distribution that coincide with the target scenario in Hiroshima 06 July 2018.

In Fig 5.17, the RMSE and RMSPE of vehicle passenger and all transport mode passenger population correspond with the population error tends. The rmse and rmspe curve looks flat and steady than experiment 2, but the absolute error is higher than experiment 2.

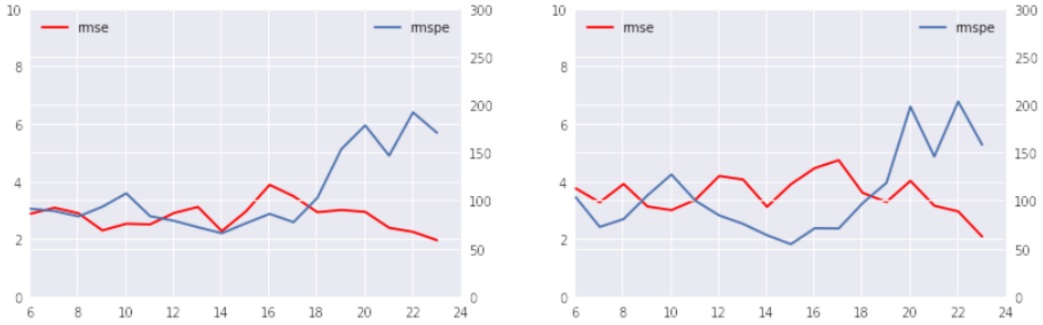

FIGURE 5.17: RMSE and RMSPE of vehicle passenger estimation result(left) and all transport mode passenger estimation result(right) in Hiroshima

5.3 Summary

In this section, we develop the framework of people mass movement simulation based on the reinforcement learning agent model developed from section 3 and section 4, and set three experiments to evaluate the simulation performance using real world GPS data collected from Yahoo! Japan. Since the agent model do not refer to any real person that contained in the dataset, it is difficult to compare the result at individual level. Instead, for both three experiments, we aggregate the synthetic trajectories and ground truth data, and use the correlation coefficient, RMSE and RMSPE of hourly population distribution and passenger distribution to evaluate the results.

In experiment 1, we set training data and ground truth data in the same day to validate the proposed agent model could reproduce realistic people mass flow at citywide level. In experiment 2, we focus on reproducing people flow on the day that rare events happened. Unlike experiment 1, we collected training data from the day that similar events have happened in the same place. In experiment 3, we further extend the application of proposed method, applying trained agent model to other area for simulation to see what people flow would like if the same event happens. Both three experiments have achieved around over 0.8 correlation comparing to ground truth.

We also noticed there are some drawbacks existing for the current model structure.

- First, the reward function in this study is represented with linear function, so some environment features such as temperature, sunshine, rainfall cannot be added to reward function even though they affected people behaviors. The results also reflect that if the environment is different between training and simulation target, the result could hardly be coincide with the ground truth.

- Second, comparing to some traditional travel demand estimation and simulation studies in transportation domain, our simulation performance is not high enough (like some traffic simulator could achieve higher than 0.9 correlation or can reproduce link based traffic volume). One reason is that such studies are based on detailed census data or manually designed agent behavior profile, the overall traffic volume or activity places are known before simulation, or statistical information such as departure time, go home time are used for simulation. Our drawback is that since we use anonymous location data for agent modeling, no personal attributes can be used for generate agent profile such as housing, income, job and age.
- Besides, proposed agent model does not use any statistical information to help behavior choices except reward function. Last, our simulation is based on the hypothesis that human mobility can be classified with specific scenario types such as weekday, holiday, heavy rain or snow, and for each scenario human mobility will not change overtime. However, this hypothesis is weak so that when we try to transfer learned agent model in a specific scenario but in different days, the simulation performance decreased badly, because there may be a lot of other factors that affect human mobility that we did not discuss.

6 Conclusion

6.1 Results and Contributions

In this thesis, we focused on individual daily movement and proposed a reinforcement learning based agent modeling and a simulation framework to reconstruct people flow on citywide level. We take individual travelers as agents and modeled their daily travel schedule as a sequential decision making by using Markov Decision Process. Unlike any previous agent-based simulation approaches, agent's behavior rules are automatically learned from the interaction with environment by reinforcement learning.

In reinforcement learning framework, agent are motivated by the feedback reward from environment, and in most successful cases such as video game, chess and robot control, the rewards are well understood and defined. However, in real world applications, especially in the case that taking human-being as agent, the reward becomes complicated and hard to define. To achieve human-like level agent control performance, we introduce inverse reinforcement learning to estimate reward function from real human-beings trajectories derived from location data. We discuss the relationship between the training data amount and estimation results. We also recover the reward function from different area in Tokyo Metropolitan area and different scenarios such as normal day and rare events(heavy snow and rain).

Furthermore, we develop the framework of people mass movement simulation framework based on developed agent model, and set three experiments to evaluate the simulation performance using real world GPS data collected from Yahoo! Japan. In experiment 1, we set training data and ground truth data in the same day to validate the proposed agent model could reproduce realistic people mass flow at citywide level. In experiment 2, we focus on reproducing people flow on the day that rare events happened. Unlike experiment 1, we collected training data from the day that similar events have happened in the same place. In experiment 3, we further extend the application of proposed method, applying trained agent model to other area for simulation to see what people flow would like if the same event happens. Both three experiments have achieved around over 0.8 correlation comparing to ground truth.

6.2 Future Directions

There are still some existing problems and future directions need to be further studied.

6.2.1 Network based environment modeling

In this study, we discretized and modeled urban environment at mesh-level, so the agents movement in space are not continuous that they simply jump from origin place to destination place. Such setting could effectively decrease the calculation cost and easily control the step (we define all the action can be done in one time step) of agents following MDP structure. However, such settings make it impossible to output the link-based traffic volume that transportation researchers and urban planners are most interested in. The difficulties of constructing a network based simulation environment using real work road and public transit network are as follows.

- The agent behaves on road network could not only make behavior decisions about destination choice and transportation mode, it need simultaneous consider the route choice which will also affect the reward from environment. However, these two decision choice making are on the two different level and current reinforcement learning do not provide any approaches to solve this issue.
- The size of network environment is much larger than mesh-based model. Take Tokyo special wards as example, there are only 1400 mesh in the area that consist of the state space. However, the road network in the same area contains more than 100,000 nodes and links, the enormous state space will make calculation not tractable and lower down the behavior accuracy.

6.2.2 Power of deep learning

There are two parts of this study that can introduce deep learning to the current structure. The one is to use deep reinforcement learning to calculate policy, the other is to use neural network to represent reward function. For the former problem, although deep reinforcement learning have achieved a lot of success in video games, chess and robot control, the application in real world is still not practical. Even the most effective algorithm, training for a robust policy needs millions or even more of episode that collected from dataset or generated from simulator are needed to update the neural network.

On the contrary, using neural network to represent reward function is a promising way to extend the application of inverse reinforcement learning. We will keep working on this topic in the future.

6.2.3 Feature engineering for inverse reinforcement learning

In this study, we introduce inverse reinforcement learning approach for solving the high dimensional complex reward function formulation issue. Currently, we leverage linear function to represent reward which has been proven practical and useful. However, we found there several limitation of using linear function for modeling travel related behavior. For example, features such as day of time, temperature and individual attributes are hard to be included into current reward function. To solve this problem, we will introduce more effective and complex formulation to represent reward function such as gaussian model and deep neural network. Furthermore, selecting feature set is another future direction.

Bibliography

[1] URL: http://www.mlit.go.jp/crd/tosiko/zpt/pdf/zenkokupt_gaiyouban_english.pdf.

[2] Pieter Abbeel and Andrew Y Ng. “Apprenticeship learning via inverse reinforcement learning”. In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 1.

[3] *About the National Land Numerical Information*. http://nlftp.mlit.go.jp/ksj-e/index.htm Accessed: 2019-07-30.

[4] Florent Altché and Arnaud de La Fortelle. “An LSTM network for highway trajectory prediction”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 353–359.

[5] Theo Arentze and Harry Timmermans. “Social networks, social interactions, and activity-travel behavior: a framework for microsimulation”. In: *Environment and Planning B: Planning and Design* 35.6 (2008), pp. 1012–1027.

[6] Yasuo Asakura and Eiji Hato. “Tracking survey for individual travel behaviour using mobile communication instruments”. In: *Transportation Research Part C: Emerging Technologies* 12.3-4 (2004), pp. 273–291.

[7] Ana LC Bazzan and Franziska Klügl. “A review on agent-based technology for traffic and transportation”. In: *The Knowledge Engineering Review* 29.3 (2014), pp. 375–403.

[8] Richard A Becker et al. "A tale of one city: Using cellular network data for urban planning". In: *IEEE Pervasive Computing* 10.4 (2011), pp. 18–26.

[9] Moshe E Ben-Akiva and John L Bowman. "Activity based travel demand model systems". In: *Equilibrium and advanced transportation modelling*. Springer, 1998, pp. 27–46.

[10] Chandra R Bhat. "A hazard-based duration model of shopping activity with nonparametric baseline specification and nonparametric control for unobserved heterogeneity". In: *Transportation Research Part B: Methodological* 30.3 (1996), pp. 189–207.

[11] Chandra R Bhat and Frank S Koppelman. "Activity-based modeling of travel demand". In: *Handbook of transportation Science*. Springer, 1999, pp. 35–61.

[12] John L Bowman. "ACTIVITY BASED TRAVEL FORECASTING1 JOHN. L. BOWMAN AND MOSHE BEN-AKIVA". In: (1996).

[13] John L Bowman and Moshe E Ben-Akiva. "Activity-based disaggregate travel demand model system with activity schedules". In: *Transportation research part a: policy and practice* 35.1 (2001), pp. 1–28.

[14] Francesco Calabrese et al. "Real-time urban monitoring using cell phones: A case study in Rome". In: *IEEE transactions on intelligent transportation systems* 12.1 (2010), pp. 141–151.

[15] Ennio Cascetta. "Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator". In: *Transportation Research Part B: Methodological* 18.4-5 (1984), pp. 289–299.

[16] Zipei Fan et al. "CityCoupling: bridging intercity human mobility". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 718–728.

[17] Sidney Feygin. “Inferring Structural Models of Travel Behavior: An Inverse Reinforcement Learning Approach”. PhD thesis. UC Berkeley, 2018.

[18] Vincent François-Lavet et al. “An introduction to deep reinforcement learning”. In: *Foundations and Trends® in Machine Learning* 11.3-4 (2018), pp. 219–354.

[19] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. “Next place prediction using mobility markov chains”. In: *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*. ACM. 2012, p. 3.

[20] Thomas F Golob and Michael G McNally. “A model of activity participation and travel interactions between household heads”. In: *Transportation Research Part B: Methodological* 31.3 (1997), pp. 177–194.

[21] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. “Understanding individual human mobility patterns”. In: *nature* 453.7196 (2008), p. 779.

[22] Hugh Gunn. “The Netherlands National Model: a review of seven years of application”. In: *International Transactions in Operational Research* 1.2 (1994), pp. 125–133.

[23] Mohammad M Hamed and Fred L Mannering. “Modeling travelers’ postwork activity involvement: toward a new methodology”. In: *Transportation science* 27.4 (1993), pp. 381–394.

[24] Davy Janssens et al. “Allocating time and location information to activity–travel patterns through reinforcement learning”. In: *Knowledge-Based Systems* 20.5 (2007), pp. 466–477.

[25] Shan Jiang, Joseph Ferreira, and Marta C Gonzalez. “Activity-based human mobility patterns inferred from mobile phone data: A case study of Singapore”. In: *IEEE Transactions on Big Data* 3.2 (2017), pp. 208–219.

[26] Shan Jiang et al. “A review of urban computing for mobile phone traces: current methods, challenges and opportunities”. In: *Proceedings of the 2nd ACM SIGKDD international workshop on Urban Computing*. ACM. 2013, p. 2.

[27] Peter Jones. *New approaches to understanding travel behaviour: the human activity approach*. University of Oxford, Transport Studies Unit, 1977.

[28] Ryuichi Kitamura. “Incorporating trip chaining into analysis of destination choice”. In: *Transportation Research Part B: Methodological* 18.1 (1984), pp. 67–81.

[29] Kris M Kitani et al. “Activity forecasting”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 201–214.

[30] Kamwoo Lee et al. “Agent-based model construction using inverse reinforcement learning”. In: *2017 Winter Simulation Conference (WSC)*. IEEE. 2017, pp. 1264–1275.

[31] Sergey Levine. “Reinforcement learning and control as probabilistic inference: Tutorial and review”. In: *arXiv preprint arXiv:1805.00909* (2018).

[32] Sergey Levine, Zoran Popovic, and Vladlen Koltun. “Nonlinear inverse reinforcement learning with gaussian processes”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 19–27.

[33] Sergey Levine et al. “End-to-end training of deep visuomotor policies”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373.

[34] Timothy P Lillicrap et al. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).

[35] Qiang Liu et al. “Predicting the next location: A recurrent model with spatial and temporal contexts”. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[36] Xin Lu, Linus Bengtsson, and Petter Holme. "Predictability of population displacement after the 2010 Haiti earthquake". In: *Proceedings of the National Academy of Sciences* 109.29 (2012), pp. 11576–11581.

[37] Wesley Mathew, Ruben Raposo, and Bruno Martins. "Predicting future locations with hidden Markov models". In: *Proceedings of the 2012 ACM conference on ubiquitous computing*. ACM. 2012, pp. 911–918.

[38] Michael G McNally. "The four step model". In: (2000).

[39] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), p. 529.

[40] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[41] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. "Deep reinforcement learning: an overview". In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2016, pp. 426–440.

[42] Andrew Y Ng, Stuart J Russell, et al. "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1. 2000, p. 2.

[43] Eric I Pas. "The effect of selected sociodemographic characteristics on daily travel-activity behavior". In: *Environment and Planning A* 16.5 (1984), pp. 571–581.

[44] Ram M Pendyala et al. "An activity-based microsimulation analysis of transportation control measures". In: *Transport Policy* 4.3 (1997), pp. 183–192.

[45] Anthony J Richardson, Elizabeth S Ampt, and Arnim H Meyburg. *Survey methods for transport planning*. Eucalyptus Press Melbourne, 1995.

[46] Stuart J Russell. "Learning agents for uncertain environments". In:

[47] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[48] Tsuna Sasaki. “Estimation of person trip patterns through Markov chains”. In: *Publication of: Traffic Flow and Transportation* ().

[49] Christian M Schneider et al. “Unravelling daily human mobility motifs”. In: *Journal of The Royal Society Interface* 10.84 (2013), p. 20130246.

[50] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[51] “Selected results of a standardised survey instrument for large-scale travel surveys in several European countries”. In: (1985).

[52] Reza Shokri et al. “Quantifying location privacy”. In: *2011 IEEE symposium on security and privacy*. IEEE. 2011, pp. 247–262.

[53] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016), p. 484.

[54] Chaoming Song et al. “Limits of predictability in human mobility”. In: *Science* 327.5968 (2010), pp. 1018–1021.

[55] Libo Song et al. “Evaluating location predictors with extensive Wi-Fi mobility data.” In: 2004.

[56] Xuan Song et al. “Modeling and probabilistic reasoning of population evacuation during large-scale disaster”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 1231–1239.

[57] Peter R Stopher and Stephen P Greaves. “Household travel surveys: Where are we going?” In: *Transportation Research Part A: Policy and Practice* 41.5 (2007), pp. 367–381.

[58] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[59] Csaba Szepesvári. “Algorithms for reinforcement learning”. In: (2009).

[60] Jameson L Toole et al. “The path most traveled: Travel demand estimation using big data resources”. In: *Transportation Research Part C: Emerging Technologies* 58 (2015), pp. 162–177.

[61] Hado Van Hasselt, Arthur Guez, and David Silver. “Deep reinforcement learning with double q-learning”. In: *Thirtieth AAAI conference on artificial intelligence*. 2016.

[62] Henk J Van Zuylen and Luis G Willumsen. “The most likely trip matrix estimated from traffic counts”. In: *Transportation Research Part B: Methodological* 14.3 (1980), pp. 281–293.

[63] Dizan Vasquez, Billy Okal, and Kai O Arras. “Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 1341–1346.

[64] Rui Wang, Xiangyu Wang, and Mi Jeong Kim. “Motivated learning agent model for distributed collaborative systems”. In: *Expert Systems with Applications* 38.2 (2011), pp. 1079–1088.

[65] Chieh-Hua Wen and Frank S Koppelman. “A conceptual and methdological framework for the generation of activity-travel patterns”. In: *Transportation* 27.1 (2000), pp. 5–23.

[66] Apichon Witayangkurn et al. “Trip reconstruction and transportation mode extraction on low data rate GPS data from mobile phone”. In:

[67] Takahiro Yabe, Kota Tsubouchi, and Yoshihide Sekimoto. “CityFlowFragility: Measuring the Fragility of People Flow in Cities to Disasters using GPS Data Collected from Smartphones”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (2017), p. 117.

[68] Min Yang et al. “Multiagent-based simulation of temporal-spatial characteristics of activity-travel patterns using interactive reinforcement learning”. In: *Mathematical Problems in Engineering* 2014 (2014).

[69] Jihang Ye, Zhe Zhu, and Hong Cheng. “What’s your next move: User activity prediction in location-based social networks”. In: *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM. 2013, pp. 171–179.

[70] Mogeng Yin et al. “A generative model of urban activities from cellular data”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.6 (2017), pp. 1682–1696.

[71] Junbo Zhang et al. “DNN-based prediction model for spatio-temporal data”. In: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM. 2016, p. 92.

[72] Junbo Zhang et al. “Predicting citywide crowd flows using deep spatio-temporal residual networks”. In: *Artificial Intelligence* 259 (2018), pp. 147–166.

[73] Yuyu Zhang et al. “Sequential click prediction for sponsored search with recurrent neural networks”. In: *Twenty-Eighth AAAI Conference on Artificial Intelligence*. 2014.

[74] Yu Zheng et al. “Mining interesting locations and travel sequences from GPS trajectories”. In: *Proceedings of the 18th international conference on World wide web*. ACM. 2009, pp. 791–800.

[75] Yu Zheng et al. “Understanding transportation modes based on GPS data for web applications”. In: *ACM Transactions on the Web (TWEB)* 4.1 (2010), p. 1.

[76] Lixin Zhou. “Active health evaluation with multi-agent”. In: *2009 Third International Conference on Genetic and Evolutionary Computing*. IEEE. 2009, pp. 169–172.

[77] Brian D Ziebart et al. “Maximum entropy inverse reinforcement learning.” In: 2008.