Human-Centric Urban Transit Evaluation and Planning

Guojun Wu[1], Yanhua Li[1], Jie Bao[2], Yu Zheng[2], Jieping Ye[3], Jun Luo[4]
[1]Worcester Polytechnic Institute, USA
[2]Urban Computing Business Unit, JD Finance
[3]Didi Chuxing
[4]Machine Intelligence Center,Lenovo Group Limited
{gwu,yli15}@wpi.edu, baojie@jd.com,msyuzheng@outlook.com
yejieping@didiglobal.com,jluo1@lenovo.com

***Abstract*—Public transits, such as buses and subway lines, offer affordable ride-sharing services and reduce the road network traffic, thus have significant impacts in mitigating the urban traffic congestion problem. However, it is non-trivial to evaluate a new transit plan, such as a new bus route or a new subway line, of its future ridership prior to actual deployment, since the travel preferences of passengers along the planned routes may vary. In this paper, we make the first attempt to model passengers' preferences of making various transit choices using Markov Decision Process (MDP). Moreover, we develop a novel inverse preference learning algorithm to infer the passengers' preferences and predict the future human behavior changes, e.g., ridership, of a new urban transit plan before its deployment. We validate our proposed framework using a unique real-world dataset (from Shenzhen, China) with three subway lines opened during the data time span. With the data collected from both before and after the transit plan deployments, Our evaluation results demonstrated that the proposed framework can predict the ridership with only 19.8% relative error, which is 23%-51% lower than other baseline approaches.**

***Index Terms*—Urban Computing, Inverse Reinforcement Learning, Human-Centric Transit Plan Evaluation**

I. Introduction

With the fast pace of global urbanization, the growth of urban population has already significantly worsen the urban traffic congestion problem [1]. By aggregating the urban trip demands with shared trains and buses, public transits offer affordable ride-sharing services and reduce the road network traffic, which in turn mitigate the traffic congestion problem. Urban public transits, such as city buses and subway passenger trains, are group travel systems, deployed for general public and operated on *established routes* and *fixed schedules*. The goal of developing new public transit plans, e.g., a new subway line or a new bus route/schedule, is to precisely meet the needs from passengers in urban areas, and attract as many passengers as possible, to take the new transit lines, from other transit modes, such as private cars. To achieve such a design goal, urban transit planners primarily conduct surveys to collect trip demand data in urban areas, and develop transit plans that cover the most trip demands. However, survey data are usually sparse and biased. Moreover, without considering the passengers' preferences (in choosing the transit modes and routes), a transit plan may lead to unexpected (too large or too small) ridership, after deploying it.

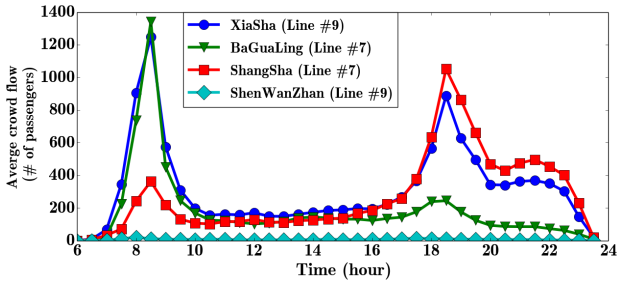


Fig. 1. Crowd flows of four new subway stations in Shenzhen, China.

Taking Shenzhen, China as an example, there were three subway lines with 63 subway stations opened in 2016. Fig 1 shows dynamic crowd flows of four subway stations along new subway lines, which count the total number of passengers going into subway stations within every half an hour. XiaSha, Baguoling, and ShangSha Station have clear diurnal patterns, with overall high crowd flows. However, at ShenWanZhan Station of Line #9, the average crowd flow is about 20 passengers per 30 minutes over the whole day. Meanwhile, from the taxi trip data, we observe numerous trips from ShenWanZhan region, both before and after Line #9 was opened. This indicates that opening the ShenWanZhan subway station did not successfully attract travelers in that region, since their travel modes did not change. With such a low ridership, it was too costly to open the ShenWanZhan subway station. In fact, it is non-trivial to evaluate the impacts of a transit deployment plan on future passenger behaviors prior to actual deployment, since the travel preferences of passengers along the planned routes may vary.

In this paper, we make the first attempt to investigate how to characterize the passenger preferences from public transit trajectory data, and develop a novel approach to predict the human behavior changes, e.g., ridership, of an urban transit plan before its deployment. Our contributions are summarized as follows.

- First, we model the travelers' trips using Markov Decision Process (MDP) model, where we consider a traveler, as an "agent", completing a trip from origin to destination by making a sequence of decisions about transport

modes and routes. Moreover, from real world data, We extract various decision-making features, that passengers evaluate when making transit choices, such as travel time, cost and level-of-convenience.
- Second, we develop a novel inverse learning algorithm to recover the preference function of passengers from their historical transit trajectories. With the passenger reward function, we develop transit plan evaluation framework to estimate the future human behaviors, i.e., ridership, crowd flow, after a transit plan is deployed.
- We validate our framework using a unique dataset from Shenzhen, China, with three subway lines opened during the data timespan. This allows us to examine our transit plan evaluation framework, with data both collected before and after the plan is deployed. Our results demonstrated that our proposed framework predicts the ridership with only 19.8% relative error from the ground-truth, which is 23%-51% lower than baseline approaches. *We will make our unique dataset available to contribute to the research community.*

The rest of the paper is organized as follows. Sec II defines the problem, describes the data and outlines our transit evaluation framework. Sec III–V introduce our proposed methodology on data-preprocessing, data-driven modeling, and transit plan evaluation. Sec VI presents evaluation results using a large-scale urban transit trajectory dataset. Finally, we conclude the paper in Sec VIII.

II. Overview

In this section, we define the transit plan evaluation problem describe datasets we use and outline the solution framework.

A. Problem Definition

In a city, its urban public transit system, including buses and subway lines, naturally forms a directed graph, for which we refer to as *public transit graph* (in short, *transit graph*) defined as follows.

Definition 1 (Transit Graph)**.** *A transit graph $G = (V, E)$ in a city represents the connections of public transit stops by the transit lines. Vertex set V is a set of transit nodes, i.e., locations of all bus stops and subway stations, and E as the set of transit edges consists of all the bus routes and subway lines between transit stops/stations.*

Passengers in an urban area are completing trip demands over time, e.g., commute trips between home and working place, where we define an urban trip demand as follows. Each passenger generates a transit trajectory, when taking the public transit system to complete a trip.

Definition 2 (Trip Demand)**.** *A trip demand td of a passenger indicates the intent of a passenger to travel from a source location src to a destination location dst from a given starting time t, which can be represented as a triple $td = \langle src, dst, t\rangle$.*

Passenger trip demands can be obtained from various data sources. For example, the transaction data from AFC devices in buses and subway systems record passenger trip demands at the level of bus stops and subway stations. Taxi GPS trajectory data with occupation information include the trip demands for taxi trips. To complete a trip demand by urban public transit, the passenger generates a transit trajectory, when traversing the transit graph.

Definition 3 (Transit Trajectory)**.** *A transit trajectory tr of a passenger trip demand $\langle src, dst, t_1\rangle$ is a sequence of spatio-temporal points, that the passenger traverses in the transit graph. Each spatio-temporal point consists of a transit node and transit edge, with a time stamp, i.e., $\ell_i = (v_i, e_i, t_i)$, where $v_i \in V$ and $e_i \in E$.*

Clearly, a spatio-temporal point $\ell = (v, e, t)$ indicates that the passenger takes a transit line $e \in E$, e.g., bus route #3, from a transit stop/station $v \in V$ at time t. With the graph representation of the urban public transit system, a new transit plan, e.g., a new subway line or a new bus route, can be represented as a graph, with a set of new transit nodes, connected by the new transit edges.

Definition 4 (Transit Plan)**.** *A transit plan $\Delta G = (\Delta V, \Delta E)$ consists of a set of new transit nodes ΔV, i.e., new bus stops and subway stations to be built, and a set of new transit edges ΔE, i.e., new bus routes and subway lines.*

Combining the transit plan graph ΔG and the existing transit graph G yields a new transit graph $G' = (V', E') = (V \cup \Delta V, E \cup \Delta E)$, which represents the urban transit connections available for the passengers, if the transit plan ΔG is deployed.

Problem Definition. Given trip demands $TD = \{td\}$ of a city, transit trajectories $TR = \{tr\}$ of passengers, existing transit graph G, and a transit plan ΔG, our goal is to evaluate/predict the ridership and crowd flow at new transit nodes $v \in \Delta V$ under G', i.e., assuming the transit plan is deployed.

B. Data Description

We use three datasets in our study, including (1) public transit trajectory data, (2) taxi trajectory data and (3) transit graph and road map data. For consistency, all these datasets are aligned with the same time period, i.e., 06/2016–12/2016.

Public transit trajectory data. In Shenzhen, all buses and subway stations are equipped with automatic fare collection (AFC) systems , where passengers swipe their smart cards at AFC devices to get aboard on a bus or enter/leave a subway station. We collected 7 months of passenger transaction data from buses and subway stations. Each transaction record contains five attributes including passenger ID, transaction type, cost, transaction time, transit station/stop name and location. The transaction type field indicates if it is an event of getting on a bus, or leaving/entering a subway station. Most of trajectories contain 0 to 3 transits and the average of transits is 1.1. These transaction data allow us to extract the trip demands and transit trajectories of passengers taking public transits. We

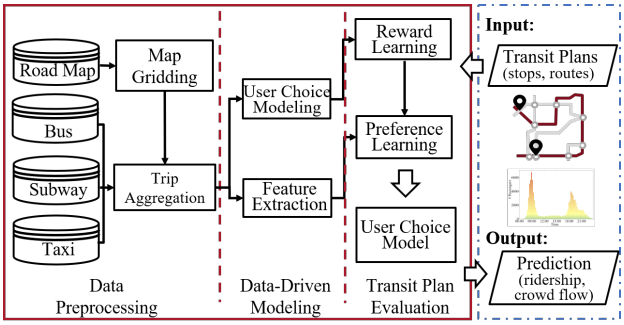


Fig. 2. Transit plan evaluation framework

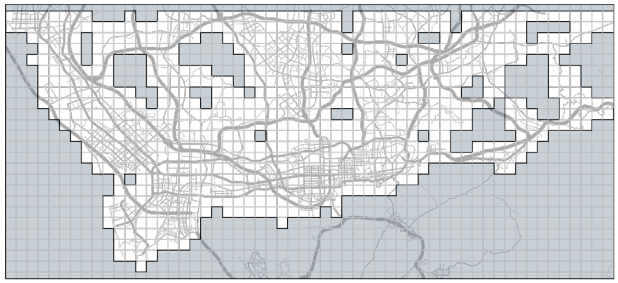


Fig. 3. Map griding

use public transit trajectory data to extract features and build the MDP transit model.

Taxi trajectory data. We collected a large-scale taxi trajectory dataset in Shenzhen. These trajectories represent $21,385$ unique taxis in Shenzhen. They are equipped with GPS sets, which periodically (i.e., roughly every 30 seconds) generate $1,797,131,540$ GPS records in total. Each record has six core attributes including unique plate ID, longitude, latitude, time, speed and passenger indicator. The passenger indicator field is a binary value for taxi data, indicating whether a passenger is on board (with value 1) or not. The taxi trip data are mainly used to extract features.

Transit Graph and Road Map Data. We retrieve a bounding box of Shenzhen city through the Google Geocoding API [2]. The bounding box was defined by latitude from $22.42°$ to $22.81°$ while longitude from $113.75°$ to $114.68°$. It covers an urban area of about 400 square miles and three million people. Within this bounding box, we obtain Shenzhen transit graph, i.e., all 892 bus routes and 8 subway lines, and road map data from OpenStreetMap [3]. This serves for feature extracting and providing connection information about bus stops and subway stations.

C. Solution Framework

Figure 2 provides an overview of our proposed framework, which consists of three main components: (i) *Stage 1 – data preparation*, which divides the urban area into equal size grids, and aggregates all the transit nodes (bus stops, subway stations), trip demands, and public transit trajectories into the grid level; (ii) *Stage 2 – data-driven modeling*, which models passenger trips as Markov decision processes, and extract various decision-making features from data; (iii) *Stage 3 – transit plan evaluation*, where we develop a novel inverse learning algorithm to learn passenger preferences and predict future ridership for a new transit plan.

III. Stage 1: Data Preprocessing

Map Griding. There are around $5,327$ bus stops and 167 subway stations in Shenzhen, China. Many of them are populated very densely, especially in downtown areas. Usually, all transit options within a certain walking distance (e.g., $500m$) are considered the same by the passengers. Hence, we partition the urban area into small regions and consider all transit stops in the same small region as a single aggregated transit stop. For the ease of implementation, we adopt the griding based method, which simply partitions the map into equal side-length grids [4], [5]. Moreover, the griding based method allows us to adjust the side-length of grids, to better examine and understand impacts of the grid size. Hence, in Stage 1, our approach divides the urban area into equal-size grids with a pre-defined side-length s in kilometers.

The remaining grid set can be represented as a graph, with grids as nodes, connected by the road network and transit system. Fig 3 highlights (in light color) those $n = 1,018$ grids covered by the road and transit network in Shenzhen, China. In Sec VI-D, we present results on evaluating the impacts of different side-lengths of grids on system performance.

Trip Aggregation. As we aggregate all transit nodes (i.e., stops and stations) into grids and each trip demand $\langle src, dst, t\rangle$ specifies a source location src, and a destination location dst, we can aggregate all trip demands to grid pairs, that is, for all trip demands with $src \in g_i$ and $dst \in g_j$, they will be considered in the same group with the source grid g_i and destination grid g_j. Similarly, we can aggregate the transit trajectories to grid level trajectories.

IV. Stage 2: Data-Driven Modeling

Passengers are making a sequence of decisions when completing a trip, such as which bus routes and subway line to take, which stop/station to transfer. Such sequential decision making processes can be naturally modeled as Markov decision processes (MDPs). Below, we will introduce some preliminaries of MDPs, and explain how we model the passenger route choice process as a MDP.

A. Markov Decision Process (MDP)

Markov decision processes (MDPs) [6] provides a mathematical framework for modeling decision making processes, where outcomes are partly random and partly under the control of a decision maker, namely, an agent. A MDP is a discrete time stochastic control process. At each time step, the process is in some state s, and the agent may choose any action a that is available at state s, where the agent receives a corresponding reward $R(s, a)$. The process responds at the next time step by randomly moving into a new state s'. The probability that the process moves into its new state s' is influenced by the

chosen action. Specifically, it is given by the state transition function $P(s' \mid s, a)$. Hence, an MDP can be represented as a 5-tuple $\langle S, A, P, R, \gamma \rangle$, where S is a finite set of states and A is a set of actions. P is the probabilistic transition function with $P(s' \mid s, a)$ as the probability of arriving at state s' by executing action a at state s, $R : S \times A \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1]$ is the discount factor, which represents the difference in importance between future rewards and present rewards. *Without loss of generality, we set $\gamma = 1$ in this work.* A randomized, memoryless policy is a function that specifies a probability distribution on the action to be executed in each state, defined as $\pi : S \times A \to [0, 1]$. The planning problem in an MDP aims to find a policy π, such that the expected total reward is maximized, namely,

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}^{\pi}(\sum_{t=0}^{T} \gamma^t R(S_t, A_t) \mid S_0 \sim \mu_0),$$

where S_t and A_t are random variables for the state and action at the time step t, and $T \in \mathbb{R} \cup \{\infty\}$ is the set of time horizons. The initial state S_0 follows the initial distribution $\mu_0 : S \to [0, 1]$. Here, Π is the memoryless policy space.

B. Modeling Transit Route Choices with MDP

We can consider each traveler in public transit system, as an "agent", who completes a trip from origin to destination by making a sequence of decisions about transit modes and routes. Inherently, each passenger evaluates various decision-making features associated with the current state and each possible decision, such as travel time, cost, level-of-convenience, by the traveler's reward function. The reward function represents the preference the traveler has over different decision-making features. Each traveler is making their decisions that maximize the total "reward" she obtains out of the trip. As a result, we model the travelers' trips (i.e., their transit route choices) using Markov Decision Process (MDP) model. Below, we explain how each component in an MDP is extracted from travelers' transit trajectory data.

Agent: Instead of viewing each individual passenger as an agent, we consider an agent as a group of passengers with nearby source and destination locations. Since in reality, people who live in the same residential community and working in the same commercial area tend to have the similar income level and family sizes, that likely lead to the similar preference profile in public transit decision making [7]. Moreover, this allows each agent (as a group of people) to have more trajectory data to learn their preference as a reward function. As we partition the entire urban area into small grids (in Stage 1), we consider all commute passenger trips with the same original and destination grids a single agent.

State set S: Each state $s \in S$ is a spatio-temporal region, denoted as a tuple $(g, \Delta t)$, where g represents a grid in the urban area and Δt is a discrete time slot with a predefined time interval. The state space S is thus finite, since the map is partitioned into a finite number of grids (e.g., $1,018$ grids in Fig 3) and each day is divided into 5-minutes intervals. For an agent, with a starting grid g_{src}, a destination grid g_{dst}, and a start time t_0, the state space only includes a limited number of spatio-temporal grids along the bus and subway lines from g_{src} to g_{dst}. For example, in Fig 4, an agent travels from grid g_{src} to g_{dst}, with two possible transfer grids (g_1 and g_2). If there are three time intervals considered in the scenarios, each grid is mapped to three MDP states, when combined with each time interval, as shown as the overlapped squares in Fig 4. Thus, there are in total 12 states in the example MDP.

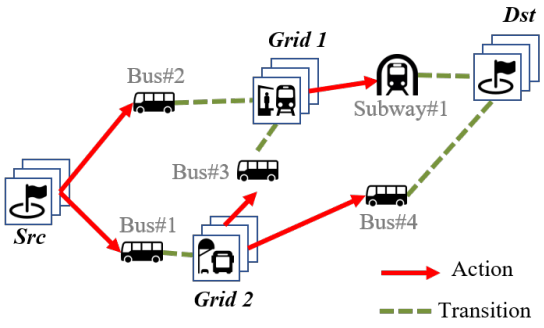


Fig. 4. Illustration: transit choices as an MDP

Action set A: An action $a \in A$ is a transit choice decision a passenger can make when completing the trip, e.g., a certain bus route or subway line with transfer stations. For example, in Fig 4, the actions the passenger can make at state g_{src} include $Bus\#2 \to g_1$ and $Bus\#1 \to g_2$.

Transition probability function $P : S \times A \times S \to [0, 1]$: Due to the dynamics of urban road traffic and crowd flow conditions, after an agent takes an action a (e.g., bus route) at a state s, the time of reaching the transferring stop may vary, leading to different state s' (of the same spatial grid but different time interval). Such uncertainty is characterized by the transition probability function as $P(s' \mid s, a)$, representing the probability of arriving at the state s' after choosing action a at the state s. The transition probability is obtained from maximum likelihood estimation from real-world urban transit trajectory data as follows. Suppose that we observed m trajectories for an agent in the historical data. Each trajectory ζ is represented as a sequence of discrete states and actions $\zeta = \{s_0, a_0, s_1, a_1, \cdots, s_N\}$ where $s_N = (g_{dst}, \Delta t_N)$ is the destination state and $s_0 = (g_{src}, \Delta t_0)$ is the source state. With this information, the maximum likelihood estimator for the transition $(s, a) \to s'$ is obtained by $P(s' \mid s, a) = \frac{N(s,a,s')}{\sum_{s' \in S} N(s,a,s')}$, where $N(s, a, s')$ is the count of this transition observed from all historical trajectories.

Reward R: When passengers make decisions of transit choices, they are considering various decision making features, such as travel time, cost, level-of-convenience. We will detail these decision-making features in Sec IV-C. In MDP, $R : S \times A \to \mathbb{R}$ represents the reward function. It captures the unique personal preferences of an agent, that maps the decision-making features (at a state s when taking an action a) to a reward value. The decision making features include travel time, cost, level-of-convenience, etc. We will discuss these features in the next subsection in more details. Such reward function $R(s, a)$ can be inversely learned from transit

trajectory data (See Sec V).

C. Decision-Making Features

Each state-action pair (s, a) is associated with a list of features, represented as a vector $\mathbf{f}_{(\mathbf{s},\mathbf{a})}$, that travelers consider when making their transit decisions. We consider three types of decision-making features, including monetary cost, time cost, level-of-convenience.

Monetary Cost indicates how much the traveler needs to spend if taking an action a at a state s, including fare and remaining cost.

•*Fare(F).* At a state-action pair (s, a), the fare feature captures the fare needed when taking an action a at a state s, e.g., the fare for taking a bus route or a subway line.

•*Remaining Cost(RC)* captures the expected additional cost needed (after taking a at s) before reaching the destination. This feature can be viewed as a measurement of how cost-efficient (s, a) is. For example, some action a taken at s, may have a smaller fare feature, but may lead to a state s', that needs at least two or more transfers (i.e., no direct transit option available at s') to reach the destination, thus incurring higher remaining cost.

Time Cost includes travel time (to the next state s'), remaining time (to meet a deadline to reach destination), and traffic speed (on road network at the current state s).

•*Travel Time(TT)* characterizes the expected travel time from the current state s to the next state after taking an action a. This feature value at a state-action pair (s, a) is estimated from the historical transit trajectory data, and is averaged over all possible next state s' from (s, a).

•*Remaining Time(RT):* We observe that most of trips in rush hours are commute trips between homes and working places. The travelers may need to meet certain deadlines to reach destinations. Such deadlines can be extracted from the historical transit trajectory data and local news. The remaining time feature captures the time difference between the current state time and the deadline, indicating how urgent the trip is.

•*Traffic Speed(TS)* indicates the average road network traffic speed in the grid and time interval of the current state s. This feature captures the potential influence to the travel time if an above-ground transit mode, e.g., a bus route, is chosen. We observe that in general a traveler has a higher chance to choose a nearby subway line if the current traffic speed is low.

Level of Convenience (LoC). Travelers may consider the comfortableness and convenience of an action a made at state s, i.e., number of transfers, number of choices, and transit mode.

•*Number of Transfers (NoT)* represents the expected number of transfers the traveler needs to take after taking action a at state s, before reaching the destination.

•*Number of Choices (NoC)* captures the total number of transit choices (including alternative bus routes, subway lines) the traveler can choose at the current state s. This feature indicates the flexibility the traveler has at a certain state s.

•*Transit Mode (TM)* is a binary indicator feature, indicating if the chosen action is a subway (with $TM = 1$) or not.

V. Stage 3: Transit Plan Evaluation

With the MDP model to characterize the decision making process of travelers, we are in a position to investigate how we may evaluate human behaviors, e.g., ridership, of a transit plan ΔG prior to its deployment. The idea is that deploying a transit plan will change the existing transit graph G to $G' = G \cup \Delta G$, thus change the MDP with updated state space $S' = S \cup \Delta S$ and action space $A' = A \cup \Delta A$. However, the unchanged element in MDP is the travelers' preferences (i.e., reward functions), namely, how they evaluate different decision-making features to make transit choices. To evaluate the human behaviors (i.e., ridership) of a transit plan ΔG, we need to answer two questions:

Q1: Given the historical transit trajectory data of an agent (collected prior to a transit plan deployment), how can we learn the preference (reward) function R of the agent, that maximizes the likelihood of the collected data being generated?

Q2: With the agents' reward functions (learned from data collected prior to the transit plan deployment), how can we predict the future human behaviors, (e.g., ridership) of a new transit plan, after it is deployed?

To answer $Q1$, we develop a novel preference learning algorithm to extract the reward functions of agents (i.e., travelers) (Sec V-A). For $Q2$, we implement a policy iteration algorithm to infer the travelers' behaviors (as MDP policies) from the updated MDP with new state and action spaces S' and A' and the extracted traveler reward function R (See Sec V-B).

A. Preference Learning

User choice modeling and preference learning has been extensively studied in literatures aiming to learn people's decision-making preferences from data they generated [8], [9], [10], [11]. Maximum-entropy inverse reinforcement learning (MEIRL) [10] considers users making a sequence of decisions in MDP, which matches our passenger decision-making process well. However, MEIRL has a strong assumption that reward functions are linear. Below, we develop a novel preference learning algorithm to capture the general non-linear reward function of travelers.

1) Passenger Preference Learning: Maximum entropy IRL assumes each passenger reward function $R(s, a)$ to be a linear function to the feature vector $\mathbf{f}_{(s,a)}$ at (s, a). Such strong assumption limits its ability to accurately learn passengers' preferences. In Sec VI, we show comparison results with real world data between linear vs non-linear reward functions. Now, we will first expand MEIRL to allow non-linear reward functions, and then develop a computational efficient algorithm to inversely learn a non-linear reward function $R(s, a)$ that best match the collected transit trajectory data.

Consider a general non-linear reward function $R(s, a)$. With the principle of maximum entropy, $P(\zeta)$, the probability of a trajectory ζ being generated, can be uniquely obtained by solving the optimization problem below.

$$\textbf{Problem} \quad \textbf{P1}: \quad \max_{P(\zeta)}: \sum_{\zeta \in TR} P(\zeta)(-\ln P(\zeta)), \quad (1)$$

$$s.t. \sum_{\zeta \in TR} P(\zeta) = 1, \quad (2)$$

$$\sum_{\zeta \in TR} P(\zeta) R(\zeta) = \tilde{R}. \quad (3)$$

$\tilde{R} = \frac{1}{|\tilde{TR}|} \sum_{\zeta \in \tilde{TR}} R(\zeta)$ represents the empirical average reward of trajectories, with $\tilde{TR}$ as the set of collected transit trajectories (i.e., the sample space). The objective function eq.(1) is the total entropy of the trajectory distribution. Constraint eq.(2) guarantees the total probability of all trajectories equals 1. Constraint eq.(3) specifies that the expected trajectory reward matches the empirical average trajectory reward $\tilde{R}$. The close-form solution to the problem is $P(\zeta) = \frac{1}{Z_R} e^{R(\zeta)}$, with $Z_R = \sum_{\zeta \in TR} e^{R(\zeta)}$, where TR is the set of all possible transit trajectories in the MDP (i.e., the population space). The proof of the solution is in the Appendix. Assume that the reward function follows a non-linear model, e.g. Neural Network, with parameter vector θ, i.e., $R(s, a, \theta)$. θ can be estimated using maximum likelihood estimation, i.e., solving the optimization problem below.

$$\max_{\theta}: \quad L(\theta) = \sum_{\zeta \in \tilde{TR}} \log P(\zeta \mid \theta) \quad (4)$$

Then, a standard gradient decent method can solve it with

$$\nabla L(\theta) = \sum_{s \in S, a \in A} \big(D^*(s, a) - D(s, a, \theta)\big) \frac{\partial R(s, a, \theta)}{\partial \theta}, \quad (5)$$

where $D^*(s, a) = \tilde{N}_{(s,a)}/|\tilde{TR}|$ and $\tilde{N}_{(s,a)}$ is the count from all transit trajectories that traverse the station-action pair (s, a). So, $D^*(s, a)$ is the empirical state-action pair visitation frequency. Similarly, $D(s, a, \theta)$ is the estimated state-action pair visitation frequency under a given reward function θ. The proof of eq.(5) is delegated to Appendix. Clearly, at i-th iteration, the state-action pair visitation frequency $D(s, a, \theta_i)$ and the partial derivative $\frac{\partial R(s,a,\theta_i)}{\partial \theta_i}$ are required when updating $\theta_{i+1} = \theta_i + \alpha \cdot \nabla L(\theta)$, where α is the step size for updating θ. As a result, this approach only works for models that allow computing $\frac{\partial R(s,a,\theta)}{\partial \theta}$, such as neural network (with backward propagation), linear regression. Models, such as decision tree, random forest, cannot be applied as reward function models, since no derivatives can be computed. Moreover, mini-batch gradient decent approach cannot be applied, since each iteration requires updating visitation frequencies for all state-action pairs. To tackle these challenges, we employ the following observation (Theorem 1) to further improve the flexibility of the non-linear preference learning algorithm. For brevity, the proof of Theorem 1 is delegated to the Appendix.

Theorem 1. *The optimization problem $P1$ is equivalent to the problem $P2$ below. They share the same optimal solution.*

$$\textbf{Problem} \quad \textbf{P2}: \max_{\theta, R^*}: \sum_{\zeta \in \tilde{TR}} \log \frac{1}{Z_R} e^{R^*(\zeta)}, \quad (6)$$

$$s.t. \sum_{\zeta \in \tilde{TR}} (R^*(\zeta) - R(\zeta, \theta))^2 \leq \epsilon, \quad (7)$$

where $R^*(\zeta)$'s are the reward values received, that best describe the demonstrated trajectory data, and $\epsilon > 0$ is a sufficiently small value.

Problem P2 can be decomposed into two subproblems as follows.

$$\textbf{Subproblem} \quad \textbf{2-1}: \quad \max_{R^*(\zeta)} \sum_{\zeta \in \tilde{TR}} \log \frac{1}{Z_R} e^{R^*(\zeta)}, \quad (8)$$

$$\textbf{Subproblem} \quad \textbf{2-2}: \quad \min_{\theta} \sum_{\zeta \in \tilde{TR}} (R^*(\zeta) - R(\zeta, \theta))^2. \quad (9)$$

From Subproblem 2-1, it is easy to prove that $R^*(s, a)$ can be learned with gradient decent, with $\nabla L(R^*(s, a)) = D^*(s, a) - D(s, a)$, and $R^*(s, a)$ represents the reward value received at a state-action pair (s, a). Then, $R^*(s, a)$ can be used as input of Subproblem 2-2 to solve an optimal model parameter θ. Here, the subproblem 2-2 can be viewed and solved using supervised learning approach, which is compatible with any supervised model. Alg 1 presents the pseudo-code of the passenger preference learning algorithm. Line 4–5 update the state-action pair visitation frequency using Alg 2. Line 6–8 update $R^*(s, a)$ with gradient decent method. Line 9 builds a supervised machine learning model using vector $\mathbf{f}_{(s,a)}$ as features, and $R^*(s, a)$ as labels. Alg 2 computes the optimal policy $\pi(s, a)$ with policy iteration, and calculates the state-action pair visitation frequency by solving a group of linear equations in eq (10) and eq (11).

$$D(s) - \sum_{s' \in S} \sum_{a' \in A(s')} D(s', a') \cdot P(s \mid s', a') = u_0(s), \forall s \in S, \quad (10)$$

$$D(s, a) = D(s)\pi(s, a), \forall (s, a) \in (S, A), \quad (11)$$

where $D(s) = \sum_{a \in A(s)} D(s, a)$ represents the state visitation frequency, namely, the likelihood of visiting the state s, if a random trajectory is generated. $u_0(s)$ is the initial distribution.

B. Transit Plan Evaluation

Given a new transit plan $\Delta G = (\Delta V, \Delta E)$, e.g., a new subway line or bus route, the new state and action spaces are $S' = S \cup \Delta S$ and $A' = A \cup \Delta A$, respectively. With the passenger preference function $R(s, a)$, the updated MDP with new transit plan is represented as $\langle S', A', P, R(s, a), \gamma \rangle$, where transition probability matrix P is considered unchanged, and discount factor $\gamma = 1$. With such a new MDP, we can apply Alg 2 to extract the optimal policies $\pi(s, a)$ passengers will employ, $D(s)$ state visitation frequency, $D(s, a)$ state-action

Algorithm 1 Preference Learning

1: **INPUT:** MDP M ,trajectories $\tilde{TR}$ and learning rate α, $\mathbf{f}_{(s,a)}$;
2: **OUTPUT:** Preference function $R^*(s,a)$;
3: Initialize $R_0^*(s,a)$; $i=1$;
4: **while** $\|\nabla L(R^*(s,a))\|_2 > \epsilon$ **do**
5: update $D(s,a)$ with $R_{i-1}^*(s,a)$ using Alg 2;
6: update $\nabla L(R^*(s,a)) = D^*(s,a) - D(s,a)$;
7: $R_i^*(s,a) = R_{i-1}^*(s,a) + \alpha * \nabla L(R^*(s,a))$;
8: $i++$;
9: Train a model $R(s,a,\theta)$ with features $\mathbf{f}_{(s,a)}$ and labels $R_{i-1}^*(s,a)$;
10: return $R^*(s,a)$, $R(s,a,\theta)$ and θ;

Algorithm 2 Computing State-Action Visitation Frequency

INPUT: $MDP = \langle S, A, P, R(s,a), 1\rangle$;
2: **OUTPUT:** State-action pair visitation frequency $D(s,a)$;
Solve optimal policy $\pi(s,a)$ from MDP with policy iteration [6];
4: Solve $D(s,a)$ from eq (10) and eq (11);
return $D(s,a)$;

pair visitation frequency. These statistics can evaluate many aspects of the new transit plan ΔG, including the ridership of new transit lines, and crowd flow at different regions over time.

Ridership of new transit lines. To evaluate the ridership of a new transit plan at a certain station, we denote (s_e, a_e) as the state-action pair of our interests. State s_e corresponds to the grid with the station to be evaluated. Recall that we consider all trips with the same source-destination grid pair as an agent. Let M denote the total number of agents in the urban area of our interests. For each agent $1 \le i \le M$, we denote n_i as the number of passengers in the agent i. we apply Alg 2 to extract the state-action pair visitation frequency $D_i(s_e, a_e)$ for agent i at the state-action pair (s_e, a_e). As a results, $Rider(s_e, a_e) = \sum_{i=1}^{M} D_i(s_e, a_e) \cdot n_i$ represents the total ridership getting on the new transit line a_e from state s_e.

Crowd flow in a grid. Similarly, to predict the total crowd flow at a grid g during certain time interval Δt, let $s_e = (g, \Delta t)$ denote the corresponding state in MDP. We calculate the state visitation frequency $D(s_e)$ over all M agents. The crowd flow at state s_e can be estimated as $Crowd(s_e) = \sum_{i=1}^{M} D_i(s_e) \cdot n_i$.

VI. Evaluations

There were three new subway lines opened between 6/01/2016 and 12/31/2016, i.e., Line #11 on June 28, 2016, Line #7 and #9 on October 28, 2016. To test our human-centric transit plan evaluation algorithm, we use data prior to the deployment to build models and predict the ridership of those new subway stations, and validate the prediction results with the data collected after their deployment.

A. Baseline methods

We conduct two sets of experiments: (i) compare reward learning (*Subproblem 2-1* in eq.(8) with other inverse learning baselines; (ii) compare ridership prediction (*Subproblem 2-2* in eq.(9) with other baseline frameworks.

Baselines for reward learning.

Our Method: Inverse Reinforcement Learning with Sub-optimal Policy(IRL+SP) (Line #1–#8 in Alg 1). It chooses the reward function with the principle of maximum entropy, and assumes that human make decisions with softmax-based suboptimal policies [10].

Baseline 1: Inverse Reinforcement Learning with Optimal Policy (IRL+OP). This baseline assumes that human decision-making follows an optimal deterministic policy, which means at each state, there is only one action being taken.

Baseline 2: Apprenticeship Learning (AL) [12]. This baseline chooses reward functions maximizing the reward gap between the best and second best policy. It assumes that passengers take optimal deterministic policy.

Baselines for ridership prediction.

Our Method: Alg 1 combined with multiple machine learning models at Line #9, such as random forest, lasso regression, and linear regression.

Baseline 1: Machine Learning Models (ML). We directly train machine learning models to predict the ridership.

Baseline 2: Multinomial Logit (MNL) Model [13]. This baseline considers each passenger make a single decision of the entire trip trajectory, rather than a sequence of decisions. MNL is used for the reward learning, where the preference learning still employs different machine learning models.

B. Experiment settings

We use stopping criteria as $\epsilon_1 = 10^{-12}$ for *Gradient Decent*, and employ the following two metrics to evaluate the reward learning and ridership prediction.

Visitation frequency difference. During the reward learning process, we not only extract reward values $R^*(s,a)$ that best match the demonstrated trajectory data, but also obtain the state-action pair visitation frequency $D(s,a)$ for each state-action pair. It can be expressed as a visitation frequency vector $D = [D(s,a)]$. Likewise, we can obtain an empirical state-action pair visitation frequency vector $\tilde{D} = [D^*(s,a)]$ from the transit trajectory data. Each $D^*(s,a)$ represents the percentage of collected trajectories that went through (s,a). The visitation frequency difference is the 2-norm difference of the two vector D and $\tilde{D}$, which characterizes how accurate the learned reward values are.

Ridership Prediction Error. Given a new subway station of a new transit plan (i.e., a subway line), which is within a grid g, denote s as a state over the grid g. The new transit line is considered as an action a. Let $N(s,a)$ be the number of passengers taking the new transit line a at state s after the new line is opened, which can be obtained from the data. Denote $N'(s,a)$ as the predicted ridership at state s, taking transit line a. The Ridership prediction error is quantified as $|N(s,a) - N'(s,a)|/N(s,a)$. The prediction $N(s,a)$ is ridership of (s,a), which is $\sum_{i=1}^{M} D_i(s,a) \cdot n_i$, the sum of ridership prediction for all related agents, where M is total

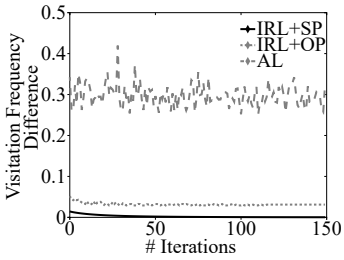


Fig. 5. Visitation frequency difference over iterations

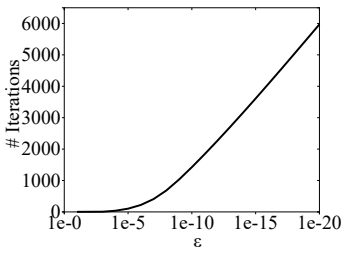


Fig. 6. Convergence rate of IRL+SP over ϵ

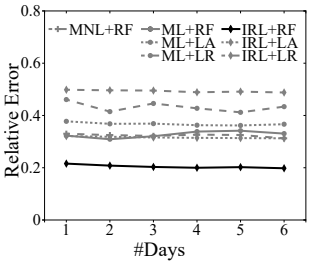


Fig. 7. Prediction error over days of data used

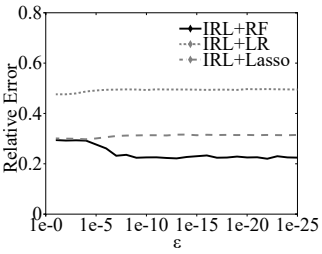


Fig. 8. Prediction error over ϵ

number of agents visiting (s, a) and n_i is the amount of passengers in agent i.

C. Preference Learning

We compare our preference learning method IRL+SP with two baseline methods, including IRL+OP and AL.

The Fig 5 clearly indicates that our IRL+SP algorithm outperforms baseline methods on visitation frequency difference. After convergence, our IRL+SP method leads to the lowest ridership vector difference, around 3×10^{-12}, where IRL+OP and AL have visitation frequency vector differences as high as 0.023 and 0.3, respectively. Such results indicate that passengers make sub-optimal decisions, rather than optimal decisions. In terms of convergence rate, Fig 5 shows that both IRL-SP and IRL-OP methods converge fast in 30 iterations, while AL method fluctuates over iterations.

Fig 6 shows how many iterations to converge with our IRL-SP algorithm (Alg 1) over stopping criteria ϵ, ranging from 10^{-1} to 10^{-20}. The results are promising: Less than 1K iterations are needed to achieve a stopping criteria of 10^{-10}.

It is also worth of mentioning that both IRL+OP and IRL+SP work in linear fashion, so their running time is similar to each other. When we set the convergence stopping criteria as $\epsilon = 3 \times 10^{-12}$. It takes about 1 minutes to converge for one agent.

D. Ridership Prediction

We evaluate the accuracy and efficiency of our proposed algorithm in predicting the ridership of a new transit plan.

Prediction accuracy. Table I shows relative errors in ridership prediction for a new transit plan, when comparing with different baselines. Clearly, our IRL-SP Algorithm (when combined with Random Forest (RF) model at Line #9 in Alg 1) yields the lowest prediction relative error, about 19.8%. IRL-SP based models (right-most column) have the lowest overall errors comparing machine learning models and multinomial logit models. This indicates that passengers are making a sequence of decisions rather than one decision when planning their trip demands. Moreover, the linear models (ML+LR, MNL and IRL+LR) shown in the first row have relative errors about 36%–49%, much higher than other non-linear models. This indicates that passengers are evaluating various decision-making features in a non-linear fashion. On the other hand, random forest based models outperform linear models and Lasso based models.

TABLE I
RIDERSHIP PREDICTION ERROR

	ML	MNL	IRL+SP
Linear	0.4337	0.3684	0.4879
Lasso	0.3665	0.3214	0.3126
Random Forest	0.3306	0.3152	0.1982

Impact of training data size. Fig 7 shows how the ridership prediction relative error changes over the amount of training data we use. We change the size of training data from 1 day data to 6 days data. The relative errors of all approaches decrease (slowly) when more training data are included.

Impact of stopping criteria ϵ. Fig 8 shows how the relative errors change with different stopping criteria ϵ of Alg 1, ranging from 10^{-1} to 10^{-25}. The relative error goes down as ϵ decreases for IRL with RF, where it keeps unchanged for IRL with LR and Lasso. Moreover, when ϵ is lower than 10^{-12}, the relative error remains stable. As a result, we choose $\epsilon = 10^{-12}$ in our experiments.

Impact of grid size. Fig 9 shows the prediction relative error changes over the grid size. We vary the grid side-length from $50m$ to $500m$, and observe that $250m$ yields the lowest relative error. This can be explained as follows: If the grid size is too small, there will be fewer stations in each grid, thus less trajectories aggregated in each grids. Such sparse data lead to high prediction error. On the other hand, with larger grids, more trajectories and stations will be aggregated, and different decision choices from passengers will be mixed together, thus increase the prediction error.

Impact of different agents. We apply all baseline algorithms to different agents, compare their prediction relative errors. Fig 10 shows the comparison results on five randomly chosen agents. It is clear that for all five agents, the orders of different baselines in prediction errors are consistent, i.e., (i) IRL based models out perform ML and MNL based models, (ii) non-linear models performs better than linear models.

E. Case studies

Now, we show three agents (group of passengers) as examples, which have significantly different reward functions. Fig 11 (A)-(C) show the locations of the source and destination grids, and their preferences to eight features.

Fig 11(A) represents travelers from Baoan to Xili. Baoan is a residential area with low housing prices, where there are many manufacture factories in Xili. So the commuters from Baoan to Xili are likely workers with low income,

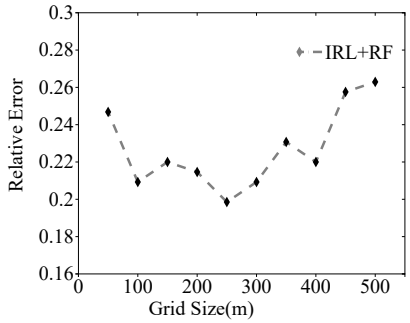


Fig. 9. Impact of grid size

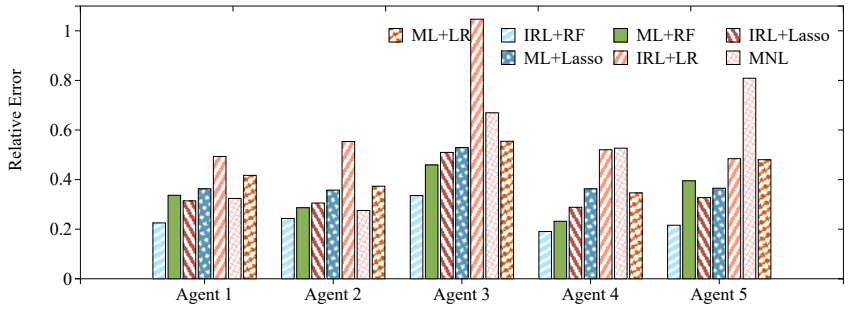
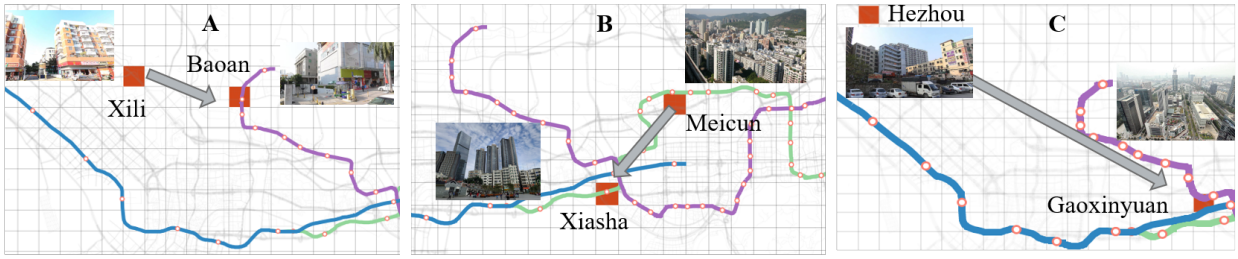


Fig. 10. Prediction error over agents



F:0.051	**RC:**0.083	**RT:**0.241	**TS:**0.090
TT:0.208	**NoT:**0.03	**NoC:**0.16	**TM:**0.017

F:0.004	**RC:**0.018	**RT:**0.089	**TS:**0.030
TT:0.681	**NoT:**0.021	**NoC:**0.131	**TM:**0.000

F:0.438	**RC:**0.000	**RT:**0.158	**TS:**0.004
TT:0.148	**NoT:**0.129	**NoC:**0.07	**TM:**0.000

Fig. 11. Illustration of 3 different agents in Shenzhen

which explains why their weights to fare and remaining cost are higher. Moreover, factories in Xili usually require their employees to follow fixed schedules. As a result, their preference to remaining time is much higher than other agents.

Fig 11(B) shows the traveler group from Meicun to Xiasha. Meicun, as a residential area, has high housing prices, where there are many technology companies, with high salary and flexible working schedule. These observations explain the preferences extracted, i.e., they weigh travel time highly while weighing the fare, remaining cost, and remaining time lower.

Fig 11(C) presents passenger group from Hezhou to Gaoxinyuan (High-Tech Park). Hezhou has lower housing price, and is in rural area away from Gaoxinyuan. Hence, passengers are more likely with low income level. This matches the preference weighing more on fare and less on travel time.

VII. Related Work

In this section, we summarize the literature works in two related areas to our study: 1) urban computing, and 2) user choice modeling.

Urban Computing. Urban computing integrates urban sensing, data management and data analytic together as a unified process to explore, analyze and solve existing critical problems in urban area such as traffic congestion, energy consumption and pollution[14]. In [15], the authors propose an dynamic urban transit system with shared shuttles using hybrid hub-and-spoke mode. The authors in [16] employ real world trajectory data of sharing bikes to develop bike lane planning strategies. In [17], [18], the authors detect urban events from heterogeneous urban datasets. However, none of those works have explicitly studied the "urban human factors", i.e., how people make decisions. Our work is the first study investigating how to quantify human preferences, and consider such preferences in urban transit planning.

User Choice Modeling. User choice modeling has been extensively studied in the literature with applications, which investigate how users make decisions in various application scenarios. For examples, In [19], they use random utility maximization and random regret minimization to analyze users' choice on park-and-ride lots. In [20], authors estimate a multinomial discrete choice model and a latent variable model of travel mode choice. In [10], the authors propose a probabilistic approach to discover reward function for which a near-optimal policy closely mimics observed behaviors. However, differing from these works, we employ data-driven approaches to study the unique decision-making process of urban public transit passengers.

VIII. Conclusion

In this paper, we introduce a novel transit plan evaluation framework that can evaluate ridership and crowd flow of a new transit plan before its deployment. In this framework, we develop a preference learning algorithm to inversely learn the passengers' preference functions when making transit decisions, and predict future human behaviors of a new transit plan. Our extensive evaluation results using real-world data demonstrated that our framework can predict the ridership with only 19.8% relative error, which is 23%-51% lower than other baseline approaches.

Acknowledgment

This work was supported by NSF CRII grant NSF CNS1657350 and a research grant from DiDi Chuxing Inc.

REFERENCES

[1] S. Wang, L. He, L. Stenneth, P. S. Yu, and Z. Li, "Citywide traffic congestion estimation with social media," in *SIGSPATIAL, 2015*, 2015.

[2] Google GeoCoding, "Road map data," https://developers.google.com/maps/documentation/geocoding/, 2016.

[3] OpenStreetMap, "Road map data," http://www.openstreetmap.org/, 2016.

[4] Y. Li, M. Steiner, J. Bao, L. Wang, and T. Zhu, "Region sampling and estimation of geosocial data with dynamic range calibration," in *ICDE, 2014*, 2014.

[5] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electric vehicles with trajectory data analytics," in *ICDE, 2015*, 2015.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[7] U. Lachapelle, L. Frank, B. E. Saelens, J. F. Sallis, and T. L. Conway, "Commuting by public transit and physical activity: where you live, where you work, and how you get there," *Journal of Physical Activity and Health*, 2011.

[8] A. A. Kumar, J. E. Kang, C. Kwon, and A. Nikolaev, "Inferring origin-destination pairs and utility-based travel preferences of shared mobility system users in a multi-modal environment," *Transportation Research Part B: Methodological*, 2016.

[9] L. Zhang, "Agent-based behavioral model of spatial learning and route choice," Tech. Rep., 2006.

[10] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *AAAI, 2008*, 2008.

[11] G. Wu, Y. Ding, Y. Li, J. Luo, F. Zhang, and J. Fu, "Data-driven inverse learning of passenger preferences in urban public transits," in *CDC, 2017*, 2017.

[12] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML, 2004*, 2004, p. 1.

[13] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, 2013.

[14] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2014.

[15] G. Liu, Y. Li, Z.-L. Zhang, J. Luo, and F. Zhang, "Citylines: Hybrid hub-and-spoke urban transit system," in *SIGSPATIAL, 2017*, 2017.

[16] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng, "Planning bike lanes based on sharing-bikes' trajectories," in *KDD, 2017*.

[17] A. Vahedian, X. Zhou, L. Tong, Y. Li, and J. Luo, "Forecasting gathering events through continuous destination prediction on big trajectory data," in *SIGSPATIAL, 2017*, 2017.

[18] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han, "Geoburst: Real-time local event detection in geo-tagged tweet streams," in *SIGIR, 2016*, 2016.

[19] B. Sharma, M. Hickman, and N. Nassir, "Park-and-ride lot choice model using random utility maximization and random regret minimization," *Transportation*, 2017.

[20] M. Paulssen, D. Temme, A. Vij, and J. L. Walker, "Values, attitudes and travel behavior: a hierarchical latent variable mixed logit model of travel mode choice," *Transportation*, 2014.

APPENDIX

A. Solution to Problem **P1**

Lemma 1. *The close-form solution for* $P1$ *(eq.(1)–(3)) is* $P(\zeta) = \frac{1}{Z_R} e^{R(\zeta)}$*, with* $Z_R = \sum_{\zeta \in TR} e^{R(\zeta)}$ *and* TR *as the set of all possible trajectories.*

Proof. Denote $P(\zeta)$ as the distribution of the trajectories generated by an agent with its optimal policy. We collected a sampled trajectory set $\tilde{TR} = \{\zeta\}$ from the agent. Problem P1 above is used to estimate the distribution $P(\zeta)$, so it matches the dataset $\tilde{TR}$ well, with a maximum entropy. The Lagrange function of P1 is represented as eq.(12).

$$L(P(\zeta)) = \sum_{\zeta \in TR} -P(\zeta) ln(P(\zeta) + \lambda_1 \big(\sum_{\zeta \in TR} P(\zeta) - 1 \big) + \lambda_2 \big(\sum_{\zeta \in TR} P(\zeta) R(\zeta) - \tilde{R} \big) \tag{12}$$

Taking the derivative of $L(P(\zeta))$ with respect to $P(\zeta)$, we obtain $\frac{\partial L(P(\zeta))}{\partial P(\zeta)} = 1 - ln(P(\zeta)) + \lambda_1 + \lambda_2 R(\zeta)$. When the derivative $\frac{\partial L(P(\zeta))}{\partial P(\zeta)} = 0$, we have $P(\zeta) = e^{1+\lambda_1} \cdot e^{\lambda_2 R(\zeta)}$. λ_2 is called "temperature" [6] used to quantify the degree to which the agent follows a sub-optimal policy vs a global optimal policy. Without loss of generality, we can set $\lambda_2 = 1$. Moreover, given the second constraint $\sum_{\zeta \in TR} P(\zeta) = 1$ (from eq.(3)), we can obtain $e^{1+\lambda_1} = 1/Z_R = 1/\sum_{\zeta \in TR} e^{R(\zeta)}$, which completes the proof. □

B. Solution to Problem in eq.(4)

Lemma 2. *The gradient to the Lagrange function in eq.(4) (with respect to* θ*) is* $\frac{\partial L(\theta)}{\partial \theta} = \sum_{s \in S, a \in A} \big(D^*(s,a) - D(s,a,\theta) \big) \frac{\partial R(s,a,\theta)}{\partial \theta}$*, where* $D^*(s,a) = \frac{\tilde{N}_{(s,a)}}{|\tilde{TR}|}$.

Proof. First of all, the Lagrange function of $P2$ is represented as $L(\theta) = \sum_{\zeta \in \tilde{TR}} \log \Big(\frac{1}{Z_R} e^{R(\zeta,\theta)} \Big)$. Taking the derivative of $L(\theta)$ with respect to θ, we can complete the following results.

$$\begin{aligned}
\frac{\partial L(\theta)}{\partial \theta} &= \frac{\partial}{\partial \theta} \sum_{\zeta' \in \tilde{TR}} \Big(R(\zeta', \theta) - \log \sum_{\zeta \in TR} e^{R(\zeta,\theta)} \Big) \\
&= |\tilde{TR}| \cdot \Big(\sum_{\zeta' \in \tilde{TR}} \frac{1}{|\tilde{TR}|} \frac{\partial R(\zeta', \theta)}{\partial \theta} - \sum_{\zeta \in TR} P(\zeta, \theta) \cdot \frac{\partial R(\zeta, \theta)}{\partial \theta} \Big) \\
&= |\tilde{TR}| \cdot \Big(\sum_{s \in S, a \in A} \big(D^*(s,a) - D(s,a,\theta) \big) \frac{\partial R(s,a,\theta)}{\partial \theta} \Big)
\end{aligned}$$

When taking the derivative equal to 0, i.e., $\frac{\partial L(\theta)}{\partial \theta} = 0$, the constant $|\tilde{TR}|$ can be removed.

□

C. Proof of Theorem 1

Proof. Denote Lagrange function of P1 and P2 as $L_1(\theta)$ and $L_2(R^*(\zeta), \theta)$, respectively. $\frac{\partial L_1(\theta)}{\partial \theta}$ is clearly eq.(5), which can be rewritten as follows, with $\tilde{P}(\zeta)$ be the empirical probability of trajectory ζ to occur.

$$\mathbf{P1}: \frac{\partial L_1(\theta)}{\partial \theta} = \sum_{\zeta \in \tilde{TR}} \big(\tilde{P}(\zeta) - P(\zeta, \theta) \big) \frac{\partial R(\zeta, \theta)}{\partial \theta} = 0.$$

Similarly, the partial derivative of $L_2(R^*(\zeta), \theta)$ is

$$\mathbf{P2}: \frac{\partial L_2(R^*(\zeta), \theta)}{\partial \theta} = \sum_{\zeta \in \tilde{TR}} \big(R^*(\zeta) - R(\zeta, \theta) \big) \frac{\partial R(\zeta, \theta)}{\partial \theta} = 0.$$

$P(\zeta, \theta) = \frac{1}{Z_R} e^{R(\zeta,\theta)}$, with $Z_R = \sum_{\zeta \in TR} e^{R(\zeta,\theta)}$, and $\tilde{P}(\zeta) = \frac{1}{\tilde{Z}} e^{R^*(\zeta)}$ with $\tilde{Z} = \sum_{\zeta \in \tilde{TR}} e^{R^*(\zeta)}$. Obviously, P2 is seeking for a $R(\zeta, \theta)$ that is equal to $R^*(\zeta)$, which means that $Z_R = \tilde{Z}$. It is thus equivalent to $P(\zeta, \theta) = \tilde{P}(\zeta)$, which is the solution from problem P1 and completes the proof. □