2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Data-Driven Inverse Learning of Passenger Preferences in Urban Public Transits

Guojun Wu, Yichen Ding, Yanhua Li, Jun Luo, Fan Zhang, Jie Fu

Abstract— **Urban public transit planning is crucial in reducing traffic congestion and enabling green transportation. However, there is no systematic way to integrate passengers' personal preferences in planning public transit routes and schedules so as to achieve high occupancy rates and efficiency gain of ride-sharing. In this paper, we take the first step tp exact passengers' preferences in planning from history public transit data. We propose a data-driven method to construct a Markov decision process model that characterizes the process of passengers making sequential public transit choices, in bus routes, subway lines, and transfer stops/stations. Using the model, we integrate softmax policy iteration into maximum entropy inverse reinforcement learning to infer the passenger's reward function from observed trajectory data. The inferred reward function will enable an urban planner to predict passengers' route planning decisions given some proposed transit plans, for example, opening a new bus route or subway line. Finally, we demonstrate the correctness and accuracy of our modeling and inference methods in a large-scale (three months) passenger-level public transit trajectory data from Shenzhen, China. Our method contributes to smart transportation design and human-centric urban planning.**

I. INTRODUCTION

In urban areas, public transit modes, such as buses and subway lines, greatly benefit the society in reducing carbon footprint and traffic congestion. However, it is challenging to design public transit routes and schedules to attract more people to use them for their (daily commute) needs. Figure 1 shows the dynamics of in-vehicle passengers of three new bus routes in Shenzhen launched in Dec 26th, 2014. It shows clearly that the two new routes M441 and M446 were popular ones, with larger numbers of passengers aboard over time, sometimes exceeding the total number of seats (the straight red line). At the same time, fewer passengers took the new M444. This indicates a potential problem in the existing planning approach. Currently, new transit plan is designed primarily based on covering the most estimated trip demands volumes [1]. This approach completely ignores the underlying passengers' personal preferences (such as waiting time, traffic condition, and so on), when selecting public transit routes.

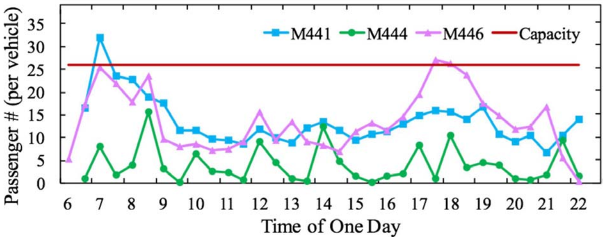

Fig. 1. Popularity of new bus routes in Shenzhen, China.

Let us consider a passenger as an agent, who decides how to reach a destination via a sequence of decisions. This is clearly a sequential decision problem, where the decisions made by the passenger depend on a certain inherent personal (reward) function of various features such as transit schedule, weather, traffic condition, etc. Hence, understanding and characterizing such personal (reward) functions of passengers will allow us to understand how passengers make decisions, which will in turn enable the urban planners to better design the public transit routes and schedules.

In the literature, various inverse reinforcement learning (IRL) methods for MDPs have been proposed to learn human behavior preferences and patterns (as reward functions) [2], [3], [4]. Apprenticeship learning algorithms [2], [3] extract the optimal reward function of an expert (e.g., a coach driver) from observed behaviors by choosing the reward function with a maximum gap of total reward from the best sub-optimal policy. However, different from the apprenticeship learning problem, passengers often adopt sub-optimal policies than a global optimal policy when choosing a public transit path to the destination. Such a problem of learning reward functions from observed sub-optimal behaviors is then related to maximum entropy Inverse Reinforcement Learning (IRL) [4], where the authors propose a probabilistic approach to discover reward function for which a near-optimal policy closely mimics observed behaviors. In recent years, this line of studies has drawn significant attentions from the research community, where various extensions and applications have been proposed, including IRL with nonlinear reward functions [5], infinite horizon decision problems [6] and learning from demonstration in robotics [7]. However, none of these works has studied the urban transit route selection problem with large-scale real data. In this paper, by analyzing a large scale (3 months) passenger-level public transit trajectory data from Shenzhen, China, we make the first attempt to extend and apply maximum entropy IRL for urban transit planning and inverse learning passengers' preferences. The contribution in this paper are three-fold: First, we construct a high-fidelity MDP model to characterize the urban transit path selection process of different passengers. Second, we develop a novel inverse learning algorithm that employs softmax policy iteration to

This work was supported in part by NSF CRII grant CNS-1657350 and a research grant from Pitney Bowes Inc.

Guojun Wu, Yichen Ding, Yanhua Li, and Jie Fu are with Worcester Polytechnic Institute (WPI), 100 Institute Road, Worcester, MA 01609, USA. {gwu,yding,yli15,jfu2}@wpi.edu

Jun Luo and Fan Zhang are with Shenzhen Institutes of Advanced Technology, Shenzhen, Guangdong 518172, China. {jun.luo,zhangfan}@siat.ac.cn

978-1-5090-2873-3/17/$31.00 ©2017 IEEE

perform gradient decent in maximum entropy IRL. Such an algorithm enables us to consider various discounting factors and different levels of sub-optimality, namely, we relate the temperature parameter in softmax policy iteration as a way to modulate the sub-optimality in agent's decision and provides deeper insight about passengers' behavior in real data. Last but not the least, we validate our method with both synthetic data and large-scale real-world urban transit data. The evaluation results demonstrate that our proposed approach can extract the passenger reward function with a near-optimal policy very close to the observed passenger behaviors, which strongly justifies our hypothesis that passenger makes sub-optimal decisions.

The rest of the paper is organized as follows. In Sec II, we briefly introduce the preliminaries of MDP and maximum entropy IRL. Sec III describes our datasets and introduces the data-driven modeling of passenger's trip trajectory selection process as an MDP. Sec IV introduces the softmax policy iteration algorithm to inversely learn the passengers' personal preferences as reward functions. Sec V evaluates the proposed approach using both synthetic data and real data. Finally, sec VI concludes the paper.

II. PRELIMINARIES

In this section, we briefly review the basics of finite Markov Decision Process and Maximum Entropy Inverse Reinforcement Learning, which are the foundations of our data-driven model and inverse learning algorithm for urban passenger preferences characterization.

A. Markov Decision Process (MDP)

An MDP is represented as a tuple $\langle S, A, P, \gamma, \mu_0, r\rangle$, 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, $\gamma \in (0, 1]$ is the discounting factor, $\mu_0 : S \to [0, 1]$ is the initial distribution, and $r : S \times A \to \mathbf{R}$ is the reward function. In our problem, each Markov Decision Process (MDP) has one terminal state $s_{\mathsf{terminal}} \in S$. It ensures that every trajectory ends at that terminal state.

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 $T \in \mathbf{R} \cup \{\infty\}$ is the set of time horizons. The initial state S_0 follows the initial distribution μ_0. Here, Π is the memoryless policy space.

B. Maximum Entropy IRL

The inverse reinforcement learning problem in MDPs aims to find a reward function $\theta : S \times A \to \mathbf{R}$ such that the distribution of action and state sequences under a (near-)optimal policy match the demonstrated behaviors. One well-known solution to Maximum Entropy IRL problem [4] proposes to find the policy, which best represents demonstrated behaviors with the highest entropy, subject to the constraint of matching feature expectations to the distribution of demonstrated behaviors.

The reward function θ is given as a linear combination of k features ϕ_i with weights θ_i such that $\forall (s, a) \in S \times A$: $r(s, a) = \theta \cdot \phi(s, a)$. And for a trajectory ρ, which is a sequence $(s_0, a_0, s_1, a_1, s_2, a_2..., s_N)$ of states $s_i \in S$ and actions $a_i \in A$, for $i = 0, \ldots, N$ where $s_N = s_{\mathsf{terminal}}$ is a terminal state in the MDP, the reward of this trajectory can be written as

$$r(\rho) = \theta \cdot \phi(\rho), \tag{1}$$

where $\phi(\rho) = \sum_{i=0}^{N-1} \gamma^i \phi(s_i, a_i)$ is the discounted feature vector counts along trajectory ρ.

Applying the principle of maximum entropy, the following equation holds,

$$\sum_{\rho\in\mathfrak{P}} P(\rho)\phi(\rho) = \tilde{\phi},$$

where $\tilde{\phi} = \frac{1}{m}\sum_{\rho\in\mathfrak{P}} \phi(\rho)$ is the expected empirical feature vector calculated from demonstrated trajectories, and $P(\rho)$ is the probability of the path ρ in the Markov chain induced from a near-optimal policy π, which ensures that if two trajectories ρ' and ρ have the same total reward, then $P(\rho) = P(\rho')$. In non-deterministic MDPs, we can easily estimate the distribution using maximum likelihood estimation, $\theta^* = \arg\max_\theta L(\theta) = \arg\max_\theta \sum_{\rho\in\mathfrak{P}} \log P(\tilde{\rho}|\theta)$. Then, a standard gradient decent method can solve it with

$$\nabla L(\theta) = \tilde{\phi} - \sum_{\rho\in\mathfrak{P}} P(\rho|\theta)\phi(\rho) = \tilde{\phi} - \sum_{i=0}^{N-1} x(s_i, a_i)\phi(s_i, a_i), \tag{2}$$

where $x(s_i, a_i) = \sum_{i=0}^{N-1} \gamma^t x_t(s_i, a_i)$ is discounted state-action (s_i, a_i) visitation frequency and $x_t(s_i, a_i)$ is state-action (s_i, a_i) visitation frequency at time step t. For infinite horizon planning, the time step N can be chosen as the mixing time in the Markov chain induced with the optimal policy π in the MDP [8].

To compute the state visitation frequency, [4] proposes an algorithm, which does not consider discounting factor γ, and the level of agent's sub-optimality in choosing policies, which are both important factors in modeling passengers' decision making process. In Sec IV, we propose an extended maximum entropy IRL algorithm that adopts softmax policy iterations to calculate the state visitation frequency, that can naturally incorporate both discounting factor γ and the agent sub-optimality level (as a temperature factor τ).

III. DATA-DRIVEN MODELING OF URBAN TRANSIT CHOICE

Now, we are in a position to elaborate on the real world datasets we use, the process we prepare the data for our study, and the MDP model we develop to capture passengers' decision making process.

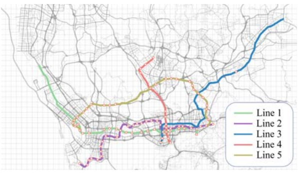

Fig. 2. Shenzhen subway lines and road map

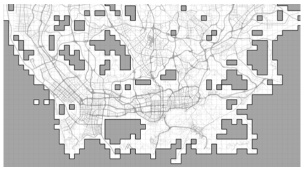

Fig. 3. Map griding ($\ell = 0.01°$)

A. Data set description

We employ two sets of urban data sources for our study, including both urban transportation infrastructure data and public transit transaction data, which are detailed as follows. **Urban transportation infrastructure data** include urban road network structure, bus routes, and subway lines in Shenzhen, China. In our study, we retrieve a bounding box of Shenzhen City, specified by the south-west and north-east corners as $(22.447203°, 113.769263°)$ and $(22.70385°, 114.33991°)$ in latitude and longitude.Figure 2 visualizes all road segments and subway lines. The public transit system consists of buses routes and subway lines. In 2014, there were in total about 890 bus routes covering all the road segments on the road map. Moreover, there were five subway lines (as shown in Figure 2).

Public transit trip trajectory data. In Shenzhen, passengers take public transits, including buses and subway lines with their smart cards, and all the fares are collected by the automatic fare collection (AFC) system. Each passenger can be uniquely identified by the card ID. Our AFC data include all events, when a passenger entering, or exiting a subway station, and getting on or off a bus. For example, an AFC data record has four fields $\langle P_{ID}, S_{ID}, t, u\rangle$, where P_{ID} is the passenger ID, S_{ID} is an unique ID indicating a subway station or a bus stop of a particular bus route, t is the event time, u is a binary variable indicating if it is an entering ($u = 1$) or exiting ($u = 0$) event. Our datasets were collected during 10/1/2014-12/31/2014 in Shenzhen, China, representing on average 11 million trip segments, equivalent to 6 million trip trajectories on a regular working day.

B. Data processing

An urban trip demand 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 $\langle src, dst, t\rangle$. 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 [9]. Moreover, this allows each agent (as a group of people) to have more trajectory data samples to learn their preferences modeled by reward functions. We partition the entire urban area into small regions, so that the commute passengers with the same home and working regions are aggregated as one agent.

For the ease of implementation, in this paper, we adopt the gridding based method, which simply partitions the map into equal side-length grids [10], [11]. Moreover, the gridding based method allows us to adjust the side-length of grids, to better examine and understand impacts of the grid size. Figure 3 shows all partition grids in the bounding rectangle region of Shenzhen, China, with side-length $\ell = 0.01°$. After removing inaccessible grids, Figure 3 highlights (in light color) those grids on the road network of Shenzhen, China. Hence, each agent is represented as a source-destination grid pair during a certain time interval (e.g., morning rush hour 7–9AM), representing all trip trajectories that start from the source grid and end at the destination grid during that time interval. In the next subsection, we develop a data-driven MDP model for each agent to characterize the decision-making problem of the trip demands of the particular agent. With the model, we will further inversely learn the agent's preference (See Sec IV).

C. Data-driven model: Urban transit choice as an MDP

Given the agents defined above, we model the process of how a passenger (of an agent) makes an urban public transit choice as an optimal planning problem i.e., an MDP

$$M = \langle S, A, P, \gamma, \mu_0, r\rangle.$$

Below, we detail how each MDP component can be extracted from real world data.

State set S: Each state $s \in S$ is spatio-temporal region, denoted as a tuple (g, t), where g represents a grid on the road map and t is a discrete time slot with a predefined time interval. The state space S is finite, since the map is partitioned into a finite number of grids (e.g., $1,018$ grids in Figure 3) and each day is divided into a fixed number of 5-minutes intervals. For an agent, with a starting grid g_s, and a destination grid g_e, and morning rush time duration $7 - 9AM$, the state space only includes a limited number of spatio-temporal regions along the bus and subway lines from g_s to g_e.

Action set A: An action $a \in A$ is the decision made by a passenger in an agent, to take a certain bus route or take a subway line.

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 transfer stop may varies, 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 ρ is represented as a sequence of discrete states and actions $\rho = \{s_0, a_0, s_1, a_1, \cdots, s_N\}$ where $s_N = s_{\text{terminal}}$ is a destination and $s_0 \in S$ is a source. With this information, the maximum likelihood estimator for the transition (s, a, 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 trajectory data.

γ and r: In the MDP, γ is the discounting factor and $r : S \times A \to \mathbf{R}$ is the reward function. Both capture the unique personal preferences of an agent, and to be inversely learned from data.

IV. INVERSE INFERENCE WITH SOFTMAX POLICY ITERATION

In this section, we first present multiple features we extracted from real data, which determine how people evaluate different transit choices. Then, we propose a novel algorithm with softmax policy iteration for obtaining the state-action visitation frequency, which is used as input to the gradient descent algorithm Eq. 2 in Sec II for computing the reward function.

A. Feature extraction

As alluded earlier, the reward of an agent taking an action a at a state s is a linear combination of a list of k features, i.e., $r(s,a) = \sum_{i=1}^{k} \theta_i \phi_i(s,a) = \theta \cdot \phi(s,a)$, with $k \geq 1$. These features are presumably what passengers evaluate strongly in reality, when making their decisions of public transit choices. We construct four such features based on common sense and available data, including fare, travel time, remaining time and transfer time.

Fare. Given a state-action pair (s,a), we calculate average amount money that passengers spend on it.

Travel Time. For a state-action pair (s,a), this feature captures the average time of passengers' getting their next state s'.

Remaining Time. When we talk about commuters, there always is a deadline associate with their commuting. For instance, in Shenzhen, people usually are required to start work at 9AM. So we calculate how many minutes left before 9AM in a state-action pair (s,a).

Transfer Time. Passengers always want to avoid too much transfering between different bus routes or subway lines. So we use average number of routes that passengers at a state-action pair (s,a) would take in the future before finishing their trip as Transfer Time. It can represent level of service at certain level, also.

B. Softmax policy iteration for computing state visitation frequency

In this section, we introduce softmax policy iteration, for calculating the visitation frequency $x(s,a)$, as input to the gradient descent algorithm for solving the reward function of a given agent. The proposed algorithm consists of two stages: First, it calculates the maximum entropy policy π. Then, it computes the state-action visitation frequency in the Markov chain induced by the policy π.

Stage 1: Softmax policy iteration.

Algorithm 1 shows the algorithm to calculate maximum entropy policy $\pi(s,a)$. The first two steps initialize a uniform starting policy $\pi_0^{\mathsf{soft}}(s,a) = \frac{1}{|A(s)|}$, with $A(s)$ as the set of actions available at the state s, and set initial values of all states to be 0. Then, starting from the initial policy $\pi_0^{\mathsf{soft}}(s,a)$, Line 6–9 performs policy improvement and evaluation.

Algorithm 1 Softmax Bellman policy

```
1:  The initial policy: $\pi_0^{\mathsf{soft}}(s,a) = \frac{1}{|A(s)|}$, $k = 0$;
2:  $V_0(s) = 0$, for all $s \in S$;
3:  Set temperature $\tau \geq 0$;
4:  while $\|V_{k+1} - V_k\| \geq \epsilon$ do
5:      for $s \in S \setminus s_{\mathsf{terminal}}$, and $a \in A(s)$ do
6:          $Q_{k+1}(s,a) = \theta \cdot \phi(s,a) + \gamma \sum_{s'\in S} P(s' \mid s,a) V_k(s')$
7:          $\pi_{k+1}^{\mathsf{soft}}(s,a) = \frac{\exp(Q_{k+1}(s,a)/\tau)}{\sum_{a\in A(s)} \exp(Q_{k+1}(s,a)/\tau)}$
8:          $V_{k+1}(s) = \sum_{a\in A(s)} Q_{k+1}(s,a) \pi_{k+1}^{\mathsf{soft}}(s,a))$
9:          $V_{k+1}(s) = 0$ for $s = s_{\mathsf{terminal}}$
10:         $k \leftarrow k+1$
11: return $\pi^{\mathsf{soft}}(s,a)$
```

Comparing to classical policy iteration method [6], the policy update step is replaced by a soft update: The probability of selecting action a from state s is proportional to a weighted exponential of the state-action value, which is computed from the policy evaluation. When a temperature parameter $\tau \to 0$, the softmax policy iteration recovers to classical policy iteration step and is equivalent to a deterministic policy $\pi : S \to A$ such that $\pi(s) = \arg\max_{a\in A(s)} Q(s,a)$. The convergence to an optimal policy using softmax policy iteration is not guaranteed. On the other hand, large value of τ causes the policy to approximate a uniformly random policy. One way to enforce convergence is to adapt τ with the step of iterations. In our case, having multiple sub-optimal policies is actually desired because human does not necessarily make the optimal decisions. Our idea is to use softmax policy iteration to perform the gradient descent step with some temperature parameters, not to solve the optimal policy.

Stage 2: State-action pair visitation frequency.

For a given policy $\pi(s,a)$, we employ the following set of linear equations to solve the state-action pair visitation frequency $x(s,a)$: $\forall s \in S$,

$$\sum_{a\in A(s)} x(s,a) - \gamma \cdot \sum_{s'\in S} \sum_{a'\in A(s')} x(s',a') \cdot P(s',a',s) = u_0(s),$$

and

$$\frac{x(s,a)}{\sum_{a'\in A(s)x} x(s,a')} = \pi^{\mathsf{soft}}(s,a),$$

where u_0 is the initial distribution, π^{soft} is the softmax policy obtained with the current parameter θ and policy iteration, and variable $x(s,a)$ can be frequency of visiting state s and taking action a. Once the set of equations is solved, we can update the feature parameter θ by following (2).

V. DATA-DRIVEN EVALUATIONS

In this section, we evaluate our proposed inverse learning algorithm for extract passengers' preference reward functions, with both synthetic data, and real-world urban public transit trajectory data.

A. Evaluation configuration

We use stopping criteria as $\epsilon_1 = 1e{-}5$ for *Softmax policy iteration* and $\epsilon_2 = 1e{-}6$ for *Gradient Decent*. Given an agent, our inverse learning algorithm can learn a reward function θ^* corresponding to sub-optimal policy π^{soft} and an expected feature count vector $\phi^{\text{soft}} = \sum_{i=1}^{N-1} x(s_i, a_i)\phi(s_i, a_i)$. We measure the difference between this expected feature count vector ϕ^{soft} and the expected feature counts from demonstrated trajectories, i.e., $\tilde{\phi} = [\frac{1}{m}\sum_{i=1}^{m}\phi(\rho_i)]$, with 2-norm difference, which is referred to as *feature difference*. Moreover, we can estimate an empirical policy from the historical trajectory data $\tilde{\pi}$, using maximum likelihood method, similar to how we estimate the transition probability function (as stated in Sec III). As a second evaluation metric, we also evaluate the difference between π^{soft} and $\tilde{\pi}$, using 1-norm difference, which is called *policy difference*.

B. Evaluations with synthetic data

We evaluation the method first with synthetic data. First, we construct an MDP M with user-specified numbers of states and actions and a randomly generated transition probability function. To obtain demonstration trajectories, we define a randomized policy π in the MDP as the expert policy and produce a set of trajectories using the Markov chain induced from M with the policy π. Since there is no features discussed in Sec IV in synthetic data, we created an artificial feature vector $\phi_s(s, a)$ for each state-action pair (s, a) as follows. Each entry in $\phi_s(s, a)$ corresponds to a state-action pair in the MDP M. Only the entry in $\phi_s(s, a)$ corresponding to state-action pair (s, a) is 1, and other entries are 0's. Next, we apply our algorithm to inverse learn, from the synthetic trajectories, a reward function. The correctness and accuracy in the learned reward function is evaluated through the comparison between a policy computed from the learned reward function and the expert policy π.

Figure 4 shows the convergence of feature difference with an MDP of 25 states and 8 actions. The number of iterations to converge is 150, though the policy difference is not convergent yet at that iteration number (See Figure 5). This is reasonable, since passengers follow sub-optimal policy in reality, and there are multiple sub-optimal policies for the same reward function. It is note that at the iteration of 150 the policy difference reaches as low as 0.08. To test the efficiency in the proposed algorithm, we generated MDPs with different sizes — M_1, M_2, M_3 — with 25 states and 8 actions, 50 states and 10 actions, 100 states and 20 actions, respectively. For these MDPs, we randomly generate $10,000$ trajectories as input data. Moreover, we generated an MDP M_4 with the same size of M_3, but $100,000$ trajectories. Figure 5) shows interesting results: As we increase the size of MDPs (from M_1 to M_3), the convergence speed decreases. This is expected, because a larger MDP has a larger feature parameter vector to learn from data. On the other hand, when we increase the trajectory set, from $10,000$ (M_3) to $100,000$ (M_4), the accuracy (in terms of the policy difference) improves, but the convergence rate becomes slower. This is also expected, since more sampled trajectories provide more accurate estimation of transition probability functions and feature count vectors.

In Figure 6, we study the impact of discounting factor to the rate of convergence using the MDP with 50 states and 10 actions. Generally, as γ decreases, error increases. It is because we use discounted state-action visiting frequency $x(s_i, a_i) = \sum_{t=0}^{\infty}\gamma^t x_t(s_i, a_i)$ which can be viewed as a weighted sum over state-action visiting frequency $x_t(s_i, a_i)$ at different time step t. If we denote ϵ_t as error term in time step t, we can easily calculate error of discounted state-action visiting frequency using $\epsilon(s_i, a_i) = \sum_{t=0}^{\infty}\gamma^t\epsilon_t(s_i, a_i)$. Clearly, when decreasing γ, $\epsilon(s_i, a_i)$ tends to contain only short term errors which can be smaller than error accumulated over a long term. For example, assume we have a state s_p that only be visited at time step 20. Then we have $\sum_{a' \in A(s_p)}\epsilon(s_p, a') = \sum_{a' \in A(s_p)}\gamma^{20}x_{20}(s_p, a')$. If γ is small enough, we can assign any policy to s_p and still have very small error in s_p.

Next, we conduct experiments to see the influence of different temperatures on the convergence. Results are shown in Figure 7. For the MDP with 50 states and 10 actions, we generate 10,000 trajectories with $\gamma = 0.7$. Figure 7 shows that error goes up when temperature is either too large or too small. If temperature is too large, a policy tends to be equal probability policy which certainly has a higher error. On the other hand, if we set a small temperature, the policy tends to be an optimal deterministic policy which has a higher error because the demonstration policy is assumed to follow the principle of maximum entropy and thus sub-optimal and randomized.

C. Evaluations with real data

In this subsection, we evaluate our model on real traffic trajectory data in Shenzhen, where we extract features in Sec IV. The MDP includes 638 states and 80 actions for an agent. Figure 8 and Figure 9 show accuracy and speed of convergence. We set $\gamma = 1$ and $\tau = 1$ for discount factor and temperature parameters. In Figure 9, we compare our method with Appreticeship Learning method and MaxEnt IRL under optimal policy. The results shows that our softmax sub-optimal method outperform those two baselines. The IRL+OP method have the worst performance and the feature different bounces around 0.25, which clearly indicates that we can't model human transit decision making process as an optimal decision making process. Besides, comparing to AL, our method would achieve better result and smoothier converge. Also, We observe that our algorithm converges very fast within 40 iterations. Moreover, within the first 10 iterations, the feature difference already converges to 10^{-5}. For policy difference, it converges within 40 iterations. The policy difference is also monotonically decreasing.

Then, we use our method with different settings in terms of discounting factor γ. Figure 10 shows that we can obtain an optimal γ which has the lowest error. Recall that Figure 6 indicates that a smaller γ always tends to have a lower accuracy. But in Figure 10, we can observe that the model achieve best performance around $\gamma = 0$. This suggests that

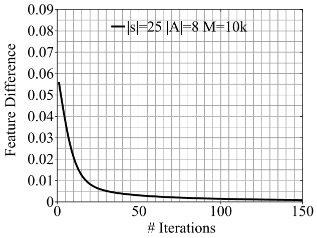

Fig. 4. Feature difference over iterations with synthetic data.

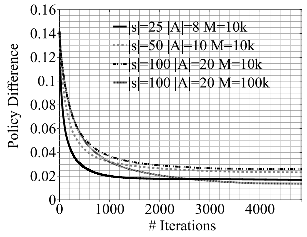

Fig. 5. Policy difference over iterations with synthetic data.

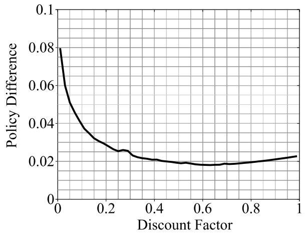

Fig. 6. Policy difference vs discounting factor γ with synthetic data.

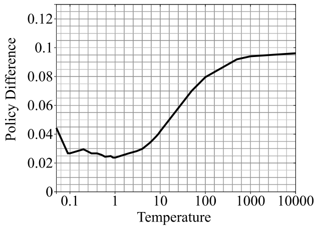

Fig. 7. Policy difference vs temperature τ with synthetic data.

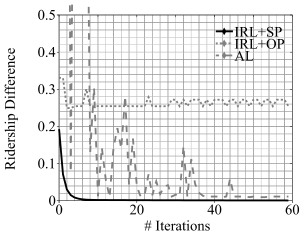

Fig. 8. Feature difference over iterations with real data.

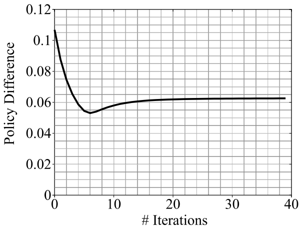

Fig. 9. Policy difference over iterations with real data.

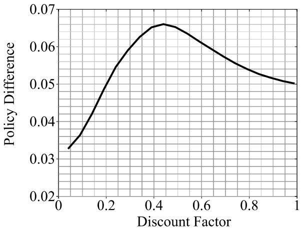

Fig. 10. Policy difference vs discounting factor γ with real data.

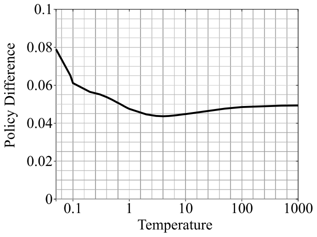

Fig. 11. Policy difference vs temperature τ with real data.

when passengers make transit decisions, they usually tends to only evaluate their choice based on current status rather than future ones. It makes sense because it is hard for passenger to predict future without limited information.

Temperature τ makes difference in agents' sub-optimal decision making too, as we show in Figure 11. Similar to Figure 7, a best τ can be found to achieve the lowest feature difference. As we have stated, higher temperature usually means that passengers tend to evaluate all possible choice as the same. The optimal τ we obtain is around 6, that implies that passengers' preference on different features we extract can play an importance role in passengers urban transit decision making.

VI. CONCLUSION

In this paper, we introduce a framework of modeling and inverse learning of human preferences in urban public transit choices. We first develop a Markov decision process model to characterize how passengers make sequential urban public transit choices. Then, we propose a novel inverse learning algorithm to extract the passengers' personal preferences, that integrates a softmax policy iteration into gradient descent in the maximum entropy IRL. This modification enables us to consider various discounting factors and different levels of sub-optimality in passengers' decision making. We conducted extensive experiments using large-scale real urban public transit data from Shenzhen, China, to evaluate our proposed method, which yielded promising results: Our proposed approach can extract the passenger reward function with near-optimal policy very close to the observed passenger behaviors, which strongly justifies our hypothesis that passenger makes sub-optimal decisions.

REFERENCES

[1] T. C. of Shenzhen Municipality, "The second batch of 2014 bus route planning in Shenzhen," http://www.sztb.gov.cn/jtzx/tzgg/201411/t20141117_5265966.htm, 2014.

[2] A. Y. Ng, S. J. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, 2000, pp. 663–670.

[3] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[4] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *23rd AAAI Conference on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conference, AAAI-08/IAAI-08*, 2008.

[5] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with gaussian processes," in *Advances in Neural Information Processing Systems*, 2011, pp. 19–27.

[6] M. Bloem and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 4911–4916.

[7] M. Herman, V. Fischer, T. Gindele, and W. Burgard, "Inverse reinforcement learning of behavioral models for online-adapting navigation strategies," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3215–3222.

[8] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Mathematical Soc., 2009.

[9] 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*, vol. 8, no. s1, pp. S72–S82, 2011.

[10] Y. Li, M. Steiner, J. Bao, L. Wang, and T. Zhu, "Region sampling and estimation of geosocial data with dynamic range calibration," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 1096–1107.

[11] 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'15: The 31st International Conference on Data Engineering*, 2015, pp. 1–12.