Inferring Passengers' Interactive Choices on Public Transits via MA-AL: Multi-Agent Apprenticeship Learning

Mingzhou Yang
ymz0228@stu.xjtu.edu.cn
Xi'an Jiaotong University
Xi'an, Shaanxi, China

Yanhua Li
yli15@wpi.edu
Worcester Polytechnic Institute
Worcester, Massachusetts

Xun Zhou
xun-zhou@uiowa.edu
The University of Iowa
Iowa City, Iowa

Hui Lu*
luhui@gzhu.edu.cn
Guangzhou University
Guangzhou, Guangdong, China

Zhihong Tian
tianzhihong@gzhu.edu.cn
Guangzhou University
Guangzhou, Guangdong, China

Jun Luo
jluo1@lenovo.com
Lenovo Machine Intelligence Center
Hong Kong

ABSTRACT

Public transports, such as subway lines and buses, offer affordable ride-sharing services and reduce the road network traffic. Extracting passengers' preferences from their public transit choices is important to city planners but technically non-trivial. When traveling by taking public transits, passengers make sequences of transit choices, and their rewards are usually influenced by other passengers' choices. This process can be modeled as a Markov Game (MG). In this paper, we make the first effort to model travelers' preferences of making transit choices using MGs. Based on the discovery that passengers usually do not change their policies, we propose novel algorithms to extract reward functions from the observed deterministic equilibrium joint policy of all agents in a general-sum MG to infer travelers' preferences. First, we assume we have the access to the entire joint policy. We characterize the set of all reward functions for which the given joint policy is a Nash equilibrium policy. In order to remove the degeneracy of the solution, we then attempt to pick reward functions so as to maximize the sum of the deviation between the the observed policy and the sub-optimal policy of each agent. This results in a skillfully solvable linear programming algorithm for the multi-agent inverse reinforcement learning (MA-IRL) problem. Then, we deal with the case where we have access to the equilibrium joint policy through a set of actual trajectories. We propose an iterative algorithm inspired by single-agent apprenticeship learning algorithms and the cyclic coordinate descent approach. We evaluate the proposed algorithms on both a simple Grid Game and a unique real-world dataset (from Shenzhen, China). Results show that when we have access to the full policy, our algorithm can efficiently recover most of the reward structure, especially the interaction of agents. In the case where we only have access to a set of sampled expert trajectories, our algorithm can provide an explanation of the expert trajectories. Measured with respect to the experts' unknown reward function, the performance of the policy output by our algorithm is close to that of the expert policy.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380235

CCS CONCEPTS

• **Theory of computation → Algorithmic game theory**; • **Computing methodologies → Modeling methodologies**.

KEYWORDS

Urban computing, Multi-Agent Apprenticeship Learning, Crowd-generated Data Mining, Markov Game

ACM Reference Format:
Mingzhou Yang, Yanhua Li, Xun Zhou, Hui Lu*, Zhihong Tian, and Jun Luo. 2020. Inferring Passengers' Interactive Choices on Public Transits via MA-AL: Multi-Agent Apprenticeship Learning. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380235

1 INTRODUCTION

With the rapid development of global urbanization, the urban traffic congestion problems have been worsen because of the growth of urban population [27]. Public transports, offering affordable ride-sharing services, contribute to reduce the road network traffic and mitigate the traffic congestion problems. Those public transits, such as bus routes and subway lines, are typically managed on schedules and operated on established routes. When planners develop new public transport plans in urban areas, they prefer these plans to meet the needs of passengers and to attract passengers to take new lines. Thus, collecting trip demand data and extracting passenger preferences from these data are important. Based on the passenger preferences, planners can efficiently predict the passengers behavior changes, e.g., ridership, caused by a public transit plan before deploying it. Then, a transit plan can lead to expected ridership after its deployment.

Traditional passenger preferences extraction methods rely on single-agent Markov Decision Process (MDP) models. Researchers consider a passenger as an "agent" which completes a trip from a point of departure to a destination by making a sequence of decisions about routes and transport modes [29][33][10]. Reward functions are learned by these techniques that explain demonstrated trajectories. However, when a traveler makes various transit choices, his/her reward may be influenced by other travelers' choices, e.g., if a lot of passengers choose to take a bus, then one

This work was done when Mingzhou Yang visited Yanhua Li's group at WPI in 2019.
*Corresponding author: luhui@gzhu.edu.cn

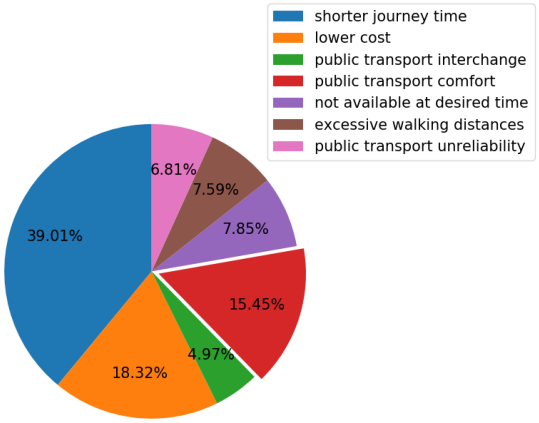

Figure 1: Reasons for preference for car rather than public transport travel for regular journeys

who do not like crowds would probably take other buses. For example, in [22], authors survey travelers' preferences in Birmingham, Burnley, Chelmsford and Reading. A result shown in Fig. 1, is that 15.45% of the travellers prefer to drive if the certain public transport mode is not comfortable (including overcrowding, seat unavailable, dirty and handling luggage [22]). Since most factors contributing to an uncomfortable journey are related to passengers' interaction, it is fair to say that modeling and analyzing passengers' interactive choices on public transits is non-trivial. As those methods based on MDPs only capture a single reward function for an agent and do not reason about competitive or cooperative motives, those inverse optimal control proves inadequate for modeling the strategic interactions of multiple agents [28].

In order to model the passengers' interaction with each other, we need to use an appropriate equilibrium solution concept as a stand-in for the notion of optimality in MDPs. Nash equilibrium [18][11], where each agent's policy is the best response to others, is a very popular solution concept for multi-agent inverse reinforcement learning (MA-IRL). Several MA-IRL algorithms based on the concept of Nash equilibrium have been proposed in recent years [28][24][13]. These algorithms can optimize agents' reward signals under the assumption that the demonstrated expert trajectories of agents are non-deterministic Nash equilibrium policies. However, we have found that passengers' choices of public transit modes are more likely to be deterministic policies. Taking the public transport dataset from Shenzhen, China as an example. This dataset contains the passengers' trajectories and their card ID from 06/2016 to 12/2016. By randomly selecting two weeks of traveling data and analyzing the trajectories of travelers who travel from a starting point to a destination for several times, we have found that more than 99% of the passengers do not alternate their choices (i.e., which transit mode to take). Therefore, it is reasonable to say that passengers' decisions can be seen as deterministic Nash equilibrium policies.

In this paper, inspired by the IRL algorithms of single-agent MDPs proposed in [19][1] and the cyclic coordinate descent approach [6][32], we propose novel algorithms to extract reward functions from deterministic Nash equilibrium joint policy. Besides, we make the first effort to investigate how to estimate the passenger preferences from real-world data by modeling the strategic interactions of them. Our contributions are summarized as follows.

- We make the first attempt to model the urban passengers' trips using MG models, where we consider a passenger, as an "agent" in an agent group, completing a trip from a starting point to a destination by making a sequence of decisions about transit modes, and its reward is influenced by the policies of other agents in the group. Moreover, from real-world data, We extract various decision-making features, that passengers evaluate when making transit mode decisions, e.g., time, cost and level-of-convenience.
- We address the MA-IRL problems in MGs. We develop a novel MA-IRL algorithm to recover the reward functions of agents from deterministic Nash equilibrium policies. Moreover, we derive a multi-agent apprenticeship learning (MA-AL) algorithm to deal with a more realistic case where the equilibrium joint policy are known only though a set of observed expert trajectories.
- We infer passengers' preferences from their interactive choices on public transits.

The rest of the paper is organized as follows. Section 2 motivates and defines the problem. Section 3 - 5 detail our proposed solution framework containing data processing, data-driven modeling and inverse preference learning. We then apply our algorithms in a simple grid world MG and a real-word preferences extraction problem in Section 6 and Section 7, respectively. Related works are discussed in Section 8. Finally, we conclude the paper and describe directions for future work in Section 9.

2 OVERVIEW

In this section, we first introduce the motivation of this study, and then formally define the problem. Finally, we give the description of the datasets and our solution framework.

2.1 Motivation

Surveys have revealed that some passengers' choices on public transits are influenced by others. Hence, extracting passengers' preferences based on a passengers' interaction model (e.g., an MG model) is likely to obtain reward functions which are closer to the real rewards than those by single-agent IRL. Moreover, with data analysis, we have found that travelers' choices on public transits are more likely to be deterministic policies rather than stochastic policies. We sampled 2 weeks, 300 pairs of departure points and destinations from passengers' public transit data in Shenzhen, China. Then, we analyzed passengers' choices when they travel between a certain starting-end point pair for several times. The result is shown in Fig. 2. The x axis in the figure represents different starting-end point pairs (e.g. Cuizhu - Grand Theatre). The y axis shows the proportion of passengers who have changed their transit choices when travelling between a certain staring-end point pair for several times. It is clear that for most starting-end point pairs (to be specifically, 89.45%), less than 10% of the passengers (denoted by the red dotted line) have ever changed their choices. By summing these proportion up, we can reveal that only 0.775%

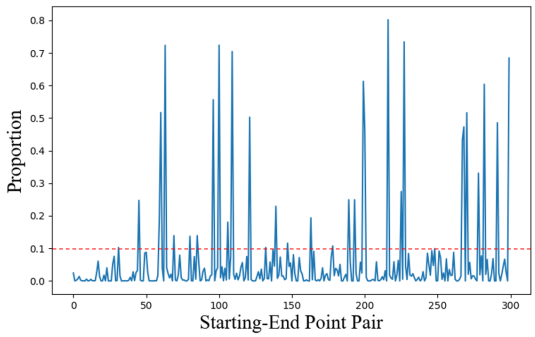

Figure 2: Proportion of passengers who have changed their choices

of the passengers have alternated their decisions. Hence, it is fair to say that passengers' policies on public transits can be seen as a Nash equilibrium joint policy. Up to now, few literature investigates how to extract agents' preferences from their deterministic policies in multi-agent area. So, we propose novel MA-AL algorithms to solve this kind of problems. Then, we can use this algorithm to infer passengers' preferences.

2.2 Problem Definition

In a city, its urban public transport system, including urban bus routes and subway lines, can be naturally seen as a directed graph. We define it as follows.

Definition 1. *(Transport Graph) [29] A transport graph $G = (\mathcal{O}; \mathcal{E})$ in a city represents the public transport stops and the transit lines connecting these stops. Vertex set $\mathcal{O}$, as a set of transit nodes, represents the locations of all bus stops and subway line stations, and $\mathcal{E}$ is a set of transit edges representing all the bus routes and subway lines connecting those stops in $\mathcal{O}$.*

In Shenzhen, there are automatic fare collection (AFC) systems in all bus stops and subway stations. Bus passengers tap their smart cards at AFC devices to get aboard, while subway passengers need to tap their cards both when they enter and leave a subway station. By collecting the transaction data from these AFC devices, we can obtain transit trajectories of each passenger. We define a transit trajectory as a sequence of spatio-temporal point $\ell = (o; e; t)$ which indicates that the passenger arrives a transit station $o \in \mathcal{O}$ at time t by a transit line $e \in \mathcal{E}$. Then we are interested in how to model the strategic interactions between passengers and then reason about their competitive and cooperative motives based on their trajectories. Intuitively, only those travelers whose trips overlapped spatially and temporally have influence on each other's decisions. Hence, we define the MA-IRL problem as follows.

Problem Definition: Given a transport graph G of a city, a departure point o_0, a destination o_T, a start time t_0 and a set of passengers who begin to travel from o_0 to o_T at t_0 and the trajectories of them, our goal is to model their interaction and then extract their preferences on transit mode choice explained by a set of features such as time, cost and level-of-convenience.

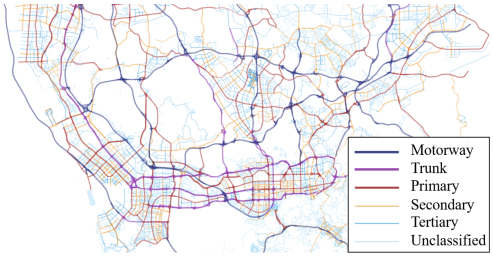

Figure 3: Shenzhen road map

2.3 Data Description

We use two datasets in our study, including public transit trajectory data (bus routes and subway lines), and transit graph data. All these datasets are adjusted to the same time period: 06/2016–12/2016.

Public transport trajectory data. We collected the passenger transaction data at AFC devices from buses and subway stations. Each record contains six attributes: card ID, transaction type, cost, record time, station name and transit mode. The transaction type indicates if the record is an event of getting aboard of a bus, or entering/leaving a subway station. The transit mode presents which transportation the passenger takes (e.g., subway line #2).

Transport Graph and Road Map Data. Taking the advantage of the Google Geocoding API [8], we used a bounding square to represent Shenzhen. The square was defined by latitude from 22.42° to 22.8°, while longitude from 113.75° to 114.68°. It covers most of the Shenzhen urban area. Within this square, we obtain Shenzhen transport graph with 892 bus routes and 8 subway lines, as well as road map data from OpenStreetMap [20], as is shown in Fig.3. This transport graph and road map data serve for feature extracting and can provide the information about bus routes and subway lines.

2.4 Solution Framework

Fig.4 provides an overview of our proposed solution framework. The framework contains three main components: (i) Stage 1 – *data processing*, which divides the urban area into equal side-length grids, and then aggregates all the transit stations into the grid level. In this stage, we also aggregate time into group. Then, by combining the aggregated time with the aggregated location data, we complete the trip aggregation; (ii) Stage 2 – *data-driven modeling*. In this stage we selects a group of passengers whose trips overlapped spatially and temporally, and model their trips as Markov Games. Besides, we extract various decision-making features from data; (iii) Stage 3 – *inverse preference learning*, where we develop a novel MA-IRL algorithm: MA-AL to learn passenger preferences from their trajectories.

3 STAGE I: DATA PROCESSING

In Shenzhen, China, there are about five thousand bus stops and more than one hundred subway stations. Many of these stations are located closely, especially in downtown areas. Usually, stations within a certain walking distance (e.g., 500m) are considered to be close. Hence, we divide the urban area into small regions. For the ease of implementation, we partition the urban area into equal side-length grids using the griding based methods [15][16]. Fig.

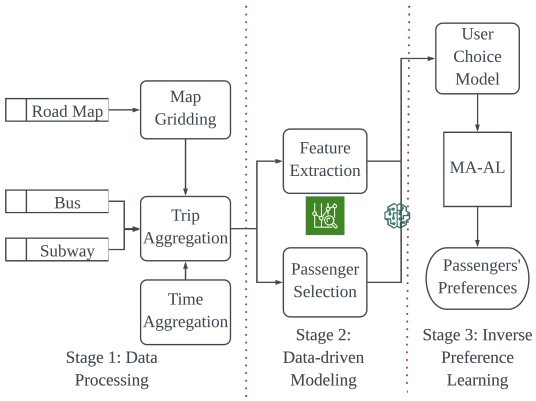

Figure 4: Solution framework

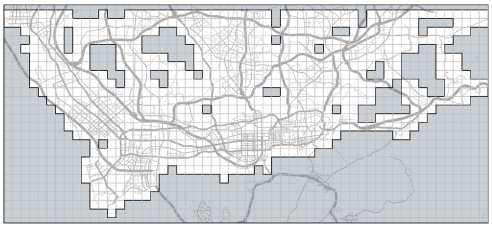

Figure 5: Map griding

5 shows the result, highlighting (in white color) those 1018 grids covered by roads and transit network in Shenzhen, China. Then, we aggregate stations in the transport graph into the grid level. Stations in the same grid are seen as a aggregated station. Similarly, we can divide a day into several equal-length (e.g., 5 minutes) time interval, and aggregate time instances to time intervals. Passengers who travel during the same time interval are seen as traveling at same time.

4 STAGE II: DATA-DRIVEN MODELING

Passengers make sequences of decisions when they complete trips, such as which bus route and subway line to take, which stop/station to transfer. By taking the interaction between passengers into consideration, such sequential decision making processes can be naturally modeled as Markov Games. Below, we will introduce some preliminaries of MGs, and explain how we model the passenger route choice process as MGs.

4.1 Markov Games

An N agents MG [7][9][24] is defined via a set of M states $\mathcal{S}$ representing the joint states of N agents, N sets of actions $\{\mathcal{A}_i\}_{i=1}^N$ and a probabilistic state transition function $P : \mathcal{S}\times\mathcal{A}_1\times\cdots\times\mathcal{A}_N \to \mathcal{P}(\mathcal{S})$ where $\mathcal{P}(\mathcal{S})$ denotes the set of probability distributions over the set $\mathcal{S}$. The probability transitions satisfy the Markov property: i.e., given that at time step t the state is s^t, the joint action of agents is $(a_1, a_2, ..., a_N)^t$ and the state transitions to s^{t+1} with probability $P(s^{t+1}|s^t, (a_1, a_2, ..., a_N)^t)$. For each agent $i = 1 : N$, a reward function is defined by $r_i : \mathcal{S} \to \mathbb{R}$. Given a discount factor $\gamma \in [0, 1)$, each agent i attempts to maximize its own total reward $R_i = \sum_{t=0}^{\infty} \gamma^t r_{i,t}$ by selecting actions based on a (stationary and Markovian) deterministic policy $\pi_i : \mathcal{S} \to \mathcal{A}_i$. The joint policy of N agents is defined by $\boldsymbol{\pi}(s) = (\pi_1(s), \pi_1(s), ..., \pi_N(s))$. Note that we use bold variables to represent the concatenation of all variables for all agents (e.g., $\boldsymbol{\pi}$ denotes joint policy : $\mathcal{S} \to \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$, $\mathbf{a}$ denotes joint actions of all agents). We use subscript $-i$ to denote all agents except for i. For example, (a_i, a_{-i}) represents the joint action of all N agents $\mathbf{a} \triangleq (a_1, a_2, ..., a_N)$.

4.2 Modeling Passengers' Interactive Choices with MG

We consider a group of passengers traveling from a same grid to a same destination at the same time. Each traveler, as an "agent", completes a trip by making a sequence of choices on public transit modes and routes. Inherently, each passenger evaluates various decision-making features (e.g., time, cost, comfort) which are associated with the current joint state of all passengers, by his/her reward function. The reward function represents the preference the passenger has over different features. We assume the reward function is linear with features in this paper. Each passenger tries to make decisions that maximize the total "reward" he/she obtains out of the trip, which leads to a Nash equilibrium joint policy. Hence, we can model the travelers' trips using an MG model. Below, we explain how each component in an MG is extracted from travelers' transit trajectory data.

Agent: We group passengers' trajectories by their starting points, destinations and departure time. Recall the data processing discussed in Section 3, we define passengers who travel from a same grid to a same grid and depart at the same time interval as a group. We are interested in extracting travelers' preferences by modelling their interaction in a group. Thus, for a specific group, the size of the group represents the number of agents in an MG and an individual passenger is an agent.

State set $\mathcal{S}$: We define the state of an agent at time step t as a spatio-temporal tuple $(g, e, \Delta t)$ which indicates that the agent is traveling to a station grid g by a transit mode e (e.g., subway line #1) and it will arrive g after time interval Δt. Hence, the state of the MG $s \in \mathcal{S}$ is the joint state of all agents. Since we have partitioned urban area into a finite number of grids and divided a day into equal minute intervals, the state set has finite states given a certain number of transit modes.

Action sets $\{\mathcal{A}_i\}_{i=1}^N$: For an agent i, an action $a \in \mathcal{A}_i$ is a transit mode choice it makes when completing a trip, e.g., a certain bus route or subway line with transfer stations. The action set $\mathcal{A}_i$ contains all possible actions of agent i.

Transition possibility function P: Due to the dynamics of urban road traffic conditions, the time it takes for an agent traveling to a station from current state: Δt may vary. Hence, after all agents choose a joint action $\mathbf{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ at a state $s^t \in \mathcal{S}$ at time step t, the state can transition to several possible states s^{t+1}, with probability $P(s^{t+1}|s^t, \mathbf{a})$. We obtain the transition function P using maximum likelihood estimation from real-world urban transit trajectories. Suppose that we observed some trajectories of a group of agents in the historical data. Each trajectory $\tau \triangleq \{(s^t, \mathbf{a}^t)\}_{t=1}^T$ denotes as a sequence of discrete states and joint

actions. Then, we estimate the transition probability $P(s'|s,\mathbf{a})$ by $P(s'|s,\mathbf{a}) = \frac{C(s,\mathbf{a},s')}{\sum_{s'\in\mathcal{S}} C(s,\mathbf{a},s')}$, where $C(s,\mathbf{a},s')$ is the count of this transition (from $(s,\mathbf{a})$ to s') in all observed trajectories.

Reward functions $\{r_i\}_{i=1}^N$: When passengers choose transit modes, they take several decision-making features into consideration, such as time cost, money consume and level-of-convenience. We assume that the reward functions of an agent r_i is linear with these features, and then we inversely learn the features weight vector of each agent. We detail these decision-making features as follow:

- *Money Cost (MC)* denotes the expected money cost from the last transfer to the current state $s \in \mathcal{S}$. This feature represents how expensive to transit to a certain state.
- *Remaining Travel Time (RTT)* denotes the expect time a passenger spends traveling from current state $s \in \mathcal{S}$ to the destination. Similar to the estimation of the transition function, we estimate the travel time using the maximum likelihood estimation from real-world data.
- *Number of Transfer (NoT)* captures the expected number of transfers a passenger needs to take before reaching the destination.
- *Number of Choices (NoC)* represents the total number of choices (on bus routes and subway lines) available at current state. This feature characterizes the flexibility the passenger has at a certain state.
- *Degree of Crowding (DoC)* characterizes how many passengers are taking the same transit mode with agent i at a certain time. We see passengers with the same tuple $(g,e,\Delta t)$ as agents in the same transportation and the number of agents in a same transportation influences the degree of crowding of every agent in this transportation.

5 STAGE III: INVERSE PREFERENCE LEARNING

In order to infer passengers' preference, we aim to derive MA-IRL algorithms for deterministic equilibrium policies. First, we deal with the case where the equilibrium policy is given.

5.1 MA-IRL from Policy

In this section, we deal with the MA-IRL problems with the joint equilibrium policy $\boldsymbol{\pi}^*$ given. Then we give the characterization of the set of the solution set of all reward functions. We then show that there are many degenerate solutions in this set and propose additional criteria to removing this degeneracy.

5.1.1 The Analogue of Bellman Equations and Bellman Optimality. An MDP is a single agent Markov game ($N = 1$). We recall Bellman's equations and Bellman optimality for a single agent MDP [25][5][19]:

Theorem 5.1 (Single-agent Bellman Equations). *Let an MDP and a policy $\pi : \mathcal{S} \to \mathcal{A}$ be given. Then, for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, the value function for the policy evaluated at any state s, $V^\pi(s)$, and the Q-function for the policy evaluated at taking action a at state s satisfy:*

$$V^\pi(s) = r(s) + \gamma \sum_{s'} P(s'|s,\pi(s))V^\pi(s') \tag{1}$$

$$Q^\pi(s,a) = r(s) + \gamma \sum_{s'} P(s'|s,a)V^\pi(s') \tag{2}$$

Theorem 5.2 (Single-agent Bellman Optimality). *Let an MDP and a policy $\pi : \mathcal{S} \to \mathcal{A}$ be given. Then π is optimal if and only if for all $s \in \mathcal{S}$:*

$$\pi(s) \in \arg\max_{a\in A} Q^\pi(s,a) \tag{3}$$

In MGs, agent i's value functions and Q-functions are defined over joint states and joint actions, rather than state-action pairs [9]. Thus, Bellman's equations can be carried over from MDPs to MGs.

Theorem 5.3 (multi-agent Bellman Equations). *Let an MG and a joint policy $\boldsymbol{\pi} : \mathcal{S} \to \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ be given. Then, for all $s \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$, for any agent i, the value function and the Q-function satisfy:*

$$V_i^{\boldsymbol{\pi}}(s) = R_i(s) + \gamma \sum_{s'} P(s'|s,\boldsymbol{\pi}(s))V_i^{\boldsymbol{\pi}}(s') \tag{4}$$

$$Q_i^{\boldsymbol{\pi}}(s,\mathbf{a}) = R_i(s) + \gamma \sum_{s'} P(s'|s,\mathbf{a})V_i^{\boldsymbol{\pi}}(s') \tag{5}$$

But the obvious analogue of theorem 5.2, in which all agents maximize their respective rewards is not adequate, because in MGs, the optimal policy of an agent depends on other agents' policies. One method is to use a multi-agent equilibrium solution concept, such as Nash equilibrium. For a normal-form general-sum game, a Nash equilibrium is defined as a fixed point where no agent can earn a higher reward by deviating its action and agents' actions are independent [18][11].

Theorem 5.4 (MG Nash Equilibrium). *Let an MG and a policy $\boldsymbol{\pi} : \mathcal{S} \to \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ be given. Then for all $s \in \mathcal{S}$, the joint policy $\boldsymbol{\pi}(s) = \mathbf{a}$ is a Nash equilibrium policy if and only if for $i = 1 : N$, for any action $\hat{a}_i \neq a_i$:*

$$Q_i^{\boldsymbol{\pi}}(s,\mathbf{a}) - Q_i^{\boldsymbol{\pi}}(s,(\hat{a}_i,a_{-i})) \geq 0 \tag{6}$$

where a_i denotes ith component of $\mathbf{a}$; $\hat{a}_i \in \mathcal{A}_i \setminus a_i$ is an alternate action of agent i; a_{-i} denotes the actions of all agents except for i.

MA-IRL Problem The MA-IRL problem for an MG is to find a state-reward function for each agent that explains the observed joint policy of agents. We consider the simple case where the model with finite state place is known and the complete policy is observed. More precisely, we are given the number of agents N, a set of states $\mathcal{S}$, N sets of agent's actions $\{\mathcal{A}_i\}_{i=1}^N$, a transition function P, a discount factor γ, and a policy $\boldsymbol{\pi}^* : \mathcal{S} \to \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$; we then wish to find reward functions of agents $\{r_i\}_{i=1}^N$ such that $\boldsymbol{\pi}^*$ is a Nash equilibrium policy in the Markov game. (We may then identify functions within this set by additional criteria.)

5.1.2 Characterization of the Solution Set. For discrete, finite space, all the functions in theorem 5.3 and 5.4 can be written as vectors or matrices. More precisely, for an MG with N agents and M states, for an agent i, $\boldsymbol{R_i}$ is an M - dimensional vector in which element j is the agent i's reward at the jth state of the MG (i.e., $r_i(j)$). Similarly, $\boldsymbol{V_i^\pi}$ is an M - dimensional vector whose element j is the value function of agent i for the joint policy $\boldsymbol{\pi}$ evaluated at state j. We also define an M - by - M matrix $\boldsymbol{P_\pi}$ in which element (i,j) gives the transition probability from state i to state j conditioned on policy $\boldsymbol{\pi}$ (i.e., $P(j|i,\boldsymbol{\pi}(i))$). Finally, we use the symbols $\geq$ and $>$ to represent

non-strict and strict vector inequality. (i.e., $\boldsymbol{x} \succ \boldsymbol{y}$ if and only if $\forall i, x_i > y_i$)

Then, our main result of the characterization of the solution set is the following:

Theorem 5.5. *Let the number of agents N, a set of states $\mathcal{S}$, N sets of agent's actions $\{\mathcal{A}_i\}_{i=1}^N$, a transition function P, a discount factor $\gamma \in [0,1)$ be given. Then the policy $\boldsymbol{\pi}^*$ is a Nash equilibrium policy if and only if, for $i = 1:N$, for all $\hat{\pi}_i^* \neq \pi_i^*$, the reward matrix of agent i $\boldsymbol{R}_i$ satisfies:*

$$(\boldsymbol{P}_{\boldsymbol{\pi}^*} - \boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)})(\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i \succeq 0 \tag{7}$$

where π_i^* denotes the ith agent's policy among the joint policy $\boldsymbol{\pi}^*$; $\hat{\pi}_i^*$ denotes an alternation policy of π_i^*; π_{-i}^* denotes the policies of all agents except i in the joint policy $\boldsymbol{\pi}^*$.

Proof. Concerning the policy $\boldsymbol{\pi}^*$, the value function equation (4) can be written as $\boldsymbol{V}_i^{\boldsymbol{\pi}^*} = \boldsymbol{R}_i + \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*}\boldsymbol{V}_i^{\boldsymbol{\pi}^*}$. Thus,

$$(\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})\boldsymbol{V}_i^{\boldsymbol{\pi}^*} = \boldsymbol{R}_i \tag{8}$$

Note that $\boldsymbol{P}_{\boldsymbol{\pi}^*}$, being a transition matrix, has all eigenvalues in the unit circle in the complex plane. With $\gamma < 1$, $\gamma \boldsymbol{P}_{\boldsymbol{\pi}^*}$ has all eigenvalues in the interior of the unit circle. Thus, the $\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*}$ has no zero eigenvalues and it is invertible. So, $\boldsymbol{V}_i^{\boldsymbol{\pi}^*}$ can be represented by:

$$\boldsymbol{V}_i^{\boldsymbol{\pi}^*} = (\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i \tag{9}$$

Substituting the Q-function equation (5) into the Nash equilibrium equation (6), we see the policy $\boldsymbol{\pi}^*$ is a Nash equilibrium policy if and only if

$$\begin{aligned}
&\sum_{s'} P(s'|s, \boldsymbol{\pi}^*(s))V_i^{\boldsymbol{\pi}^*}(s') \geq \sum_{s'} P(s'|s, (\hat{\pi}_i^*(s), \pi_{-i}^*(s)))V_i^{\boldsymbol{\pi}^*}(s'), \\
&\qquad\qquad \forall s \in S, \forall i \in [1,N], \forall \hat{\pi}_i^* \neq \pi_i^* \\
&\iff \boldsymbol{P}_{\boldsymbol{\pi}^*}\boldsymbol{V}_i^{\boldsymbol{\pi}^*} \geq \boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)}\boldsymbol{V}_i^{\boldsymbol{\pi}^*}, \quad \forall i \in [1,N], \forall \hat{\pi}_i^* \neq \pi_i^* \\
&\iff \boldsymbol{P}_{\boldsymbol{\pi}^*}(\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i \geq \boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)}(\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i, \\
&\qquad\qquad \forall i \in [1,N], \forall \hat{\pi}_i^* \neq \pi_i^*
\end{aligned}$$

where the last line in this derivation used the equation (9). This completes the proof. □

REMARK. By changing all inequalities in the proof above to strict inequalities, we can easily proof that $(\boldsymbol{P}_{\boldsymbol{\pi}^*} - \boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)})(\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i \succ 0$ is the necessary and sufficient condition for $\boldsymbol{\pi}^*$ to be the unique optimal policy.

5.1.3 LP Formulation and Penalty Terms. For most Markov games, there are many choices of $\boldsymbol{R}_i$ that meet the criteria in equation (7). For example, $\boldsymbol{R}_i = 0$ is always a solution. To remove this degeneracy, one natural way to choose $\boldsymbol{R}_i$ is to choose one so as to maximize the sum of the deviation from the quality of the equilibrium policy to the quality of the sub-optimal policy for all agents, i.e., to maximize

$$\sum_{i \in [1,N]} \sum_{s \in \mathcal{S}} (Q_i^{\boldsymbol{\pi}^*}(s, \boldsymbol{\pi}^*(s)) - \max_{\hat{\pi}_i^* \neq \pi_i^*} Q_i^{\boldsymbol{\pi}^*}(s, (\hat{\pi}_i^*(s), \pi_{-i}^*(s)))) \tag{10}$$

In addition, we optionally add a weight decay-like penalty term such as $-\lambda \|\boldsymbol{R}_i\|_1$ to the objective function to obtain a simpler reward. Then, our optimization problem is:

$$\begin{aligned}
\textit{maximize} \quad & \sum_{i=1}^{N}(\sum_{j=1}^{M} \min_{\hat{\pi}_i^* \neq \pi_i^*} \{(\boldsymbol{P}_{\boldsymbol{\pi}^*}(j) - \boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)}(j)) \\
& \qquad (\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i\} - \lambda \|\boldsymbol{R}_i\|_1) \\
s.t. \quad & (\boldsymbol{P}_{\boldsymbol{\pi}^*} - \boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)})(\boldsymbol{I} - \gamma \boldsymbol{P}_{\boldsymbol{\pi}^*})^{-1}\boldsymbol{R}_i \succeq 0, \\
& \qquad \forall i \in [1,N], \forall \hat{\pi}_i^* \neq \pi_i^* \\
& |\boldsymbol{R}_i(j)| \leq R_{max}, \quad \forall i \in [1,N], \forall j \in [1,M]
\end{aligned}$$

where $\boldsymbol{P}_{\boldsymbol{\pi}^*}(j)$ denotes the jth row of $\boldsymbol{P}_{\boldsymbol{\pi}^*}$. Similarly, $\boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)}(j)$ denotes the jth row of $\boldsymbol{P}_{(\hat{\pi}_i^*, \pi_{-i}^*)}$ and $\boldsymbol{R}_i(j)$ denotes the jth component of $\boldsymbol{R}_i$. It is fair to say that this may be formulated as a linear program and then solved efficiently.

5.2 MA-AL from Sampled Trajectories

In this section, we address the MA-IRL problems for a more realistic case where we only have access to the joint equilibrium policy $\boldsymbol{\pi}^*$ through a set of actual trajectories in the state space - i.e., a set of expert trajectories $\{\tau_E\} = \left\{\{(s^t, \mathbf{a}^t)\}_{t=1}^T\right\}$ collected by sampling under policy $\boldsymbol{\pi}^*$. To simplify notation, we assume that there is only one fix start state s^0, and that the reward functions are linear with the outcome features. For an MG with K features, we express the reward function of agent i at state s as : $r_i(s) = \boldsymbol{\theta}_i(s)\boldsymbol{\omega}_i$, where $\boldsymbol{\theta}_i : \mathcal{S} \to \mathbb{R}^K$ is a known, bounded function of features, and the K-dimensional vector $\boldsymbol{\omega}_i$ contains the reward weights which we want to fit. Under this assumption, if the set of expert trajectories contains only one trajectory : τ_E, and this sampled trajectory under $\boldsymbol{\pi}^*$ visited the sequence of states $(s^0, s^1, ..., s^T)$, then the value function of an agent i at state s^0 under the joint equilibrium policy $\boldsymbol{\pi}^*$ can be expressed as a linear function of $\boldsymbol{\omega}_i$:

$$V_i^{\boldsymbol{\pi}^*}(s^0) = (\boldsymbol{\theta}_i(s^0) + \gamma \boldsymbol{\theta}_i(s^1) + ... + \gamma^T \boldsymbol{\theta}_i(s^T))\boldsymbol{\omega}_i \tag{11}$$

If the set of expert trajectories contains several trajectories, then $V_i^{\boldsymbol{\pi}^*}(s^0)$ can be expressed as the average empirical returns of these trajectories (i.e. the average of equation 11).

Recalling the theorem of MG Nash equilibrium policy discussed in section 4.1, we want to find a reward weight vector $\boldsymbol{\omega}_i$ for each agent i so that the value function (hopefully) satisfies:

$$V_i^{\boldsymbol{\pi}^*}(s^0) \geq V_i^{(\hat{\pi}_i^*, \pi_{-i}^*)}(s^0) \tag{12}$$

where $\boldsymbol{\pi}^*$ denotes the Nash equilibrium joint policy; π_i^* denotes the ith agent's policy among the joint policy $\boldsymbol{\pi}^*$; $\hat{\pi}_i^*$ denotes an alternation policy of π_i^*; π_{-i}^* denotes the policies of all agents except i in the joint policy $\boldsymbol{\pi}^*$.

Inspired by the single-agent apprenticeship learning algorithm proposed in [1], we can use an iterative algorithm to find a reward function for agent i given the equilibrium policies of other agents π_{-i}^*. For each iteration, we solve the quadratic programming formulation below to estimate the reward function being optimized by the expert:

$$\begin{aligned}
&\max_{\boldsymbol{\omega}_i} (V_i^{\boldsymbol{\pi}^*}(s^0) - V_i^{(\bar{\pi}_i, \pi_{-i}^*)}(s^0)) \\
&s.t. ||\boldsymbol{\omega}_i||_2 \leq 1
\end{aligned}$$

where $\bar{\pi}_i$ is an estimation of the optimal policy of agent i.

Algorithm 1 Multi-Agent Apprenticeship Learning from a Sampled Trajectory

Input:
A Markov Game; A set of expert trajectories $\{\tau_E\}$;
Output:
Reward weight vectors for all agents $\{\boldsymbol{\omega}_i\}_{i=1}^N$;

Randomly initialize the estimation of equilibrium policy $\bar{\boldsymbol{\pi}}$;
repeat
  for each $i \in [1, N]$ **do**
    For each expert trajectory, calculate an empirical return using equation 11;
    Define $V_i^{\boldsymbol{\pi}^*}(s^0)$ to be the average empirical return of these expert trajectories;
    Randomly choose a policy π_i' for agent i;
    repeat
      Define $\boldsymbol{\pi}' = (\pi_i', \bar{\pi}_{-i})$;
      Sample a number of trajectories under $\boldsymbol{\pi}'$ from the initial state : s^0;
      For each trajectory, calculate an empirical return using equation 11;
      Define $V_i^{\boldsymbol{\pi}'}(s^0)$ to be the average empirical return of these trajectories;
      Solve the quadratic programming formulation :

$$\max_{\boldsymbol{\omega}_i}(V_i^{\boldsymbol{\pi}^*}(s^0) - V_i^{\boldsymbol{\pi}'}(s^0))$$
$$s.t. ||\boldsymbol{\omega}_i||_2 \leq 1$$

      Based on $\boldsymbol{\omega}_i$ and $\bar{\pi}_{-i}$, calculate the optimal policy for agent i and update π_i';
    until The quadratic programming convergence
    Update $\bar{\boldsymbol{\pi}}(s) = (\pi_i'(s), \bar{\pi}_{-i}(s))$;
  end for
until Convergence or the termination condition is reached
return $\{\boldsymbol{\omega}_i\}_{i=1}^N$;

Then, we update the estimated policy $\bar{\pi}_i$ under $\boldsymbol{\omega}_i$. When the loop converges, (i.e., $\max_{\boldsymbol{\omega}_i}(V_i^{\boldsymbol{\pi}^*}(s^0) - V_i^{(\bar{\pi}_i, \pi_{-i}^*)}(s^0)) \leq \epsilon$), which means that the estimated joint policy's performance is close to the performance of the equilibrium joint policy, we find a reward function for agent i that explains its trajectory.

Difficulties arise, however, because the Nash equilibrium joint policy of all states is not known in our setting. Instead, we only have access to $\boldsymbol{\pi}^*$ from a set of demonstrated trajectories. Thus, inspired by [32], we settle for potentially non-optimal policies by employing a cyclic coordinate descent approach. It iteratively simulates a policy for agent $i : \pi_i'$ whose performance is close to the performance of the equilibrium joint policy subject to the estimation of the joint equilibrium policy of $-i : \bar{\pi}_{-i}$.

We assume the ability to find an optimal policy for an agent i given its reward function and the policies of $-i$. Besides, we also assume that we have the ability to simulate trajectories in the MG (from an initial state) under a given joint policy. Then, we summarize the algorithm in Algorithm 1.

6 EXPERIMENTS

In our first experiment, we use a 2×2 grid game with two agents and one goal [9]. Fig.6 depicts the initial state of the game. We name the lower-left square, the lower-right square, the upper-left square and the upper-right square as square 0, 1, 2 and 3, respectively. Then the initial state of agent A is square 0, and the initial state of B is square 3. The goal for both agents is square 2. In the grid game, the action set of each agent includes one step in any of the four compass directions. Actions are executed together. The reward of each agent is the points it scores at a state. If both agents attempt to reach the same square, they both lose ten points. If ever an agent moves into the goal, it scores five points and the game is over. In this grid game, there exists several deterministic Nash equilibrium policies for both agents. For example, an equilibrium policy is that agent A moves up to square 2 and agent B moves down to square 1. Then agent A scores five and agent B scores zero, and it is an Nash equilibrium because no agent can obtain a higher score by deviating its action. By maximizing the reward of agent A, we obtained an equilibrium joint policy of (A,B) shown in Fig.7. And the true reward functions of both agents are shown in Fig.8. The inverse equilibrium problem is that of recovering the reward functions of both agents given the equilibrium joint policy and problem dynamics. Running the algorithm described in section 5.1.3, we obtained the reward functions of both agents shown in Fig.9. It has clearly recovered most of the reward structure, especially when both agents attempt to reach the same square, (i.e. the diagonal in the figure), which the algorithms of single agent IRL can never recover.

Our next experiment run on a more challenging problem: we apply the sample-based algorithm to the grid game described above. First, we randomly sample a set of expert trajectories based on the Nash equilibrium joint policy in Fig.7. All the trajectories start from the initial state shown in Fig.6 (with A in square 0 and B in square 3) and end when an agent moves to the goal (square 2). Then we use Algorithm 1 in Section 5.2 to extract reward functions of both agents from these trajectories. For the convenience of implementation, we take the advantage of the idea of the simplification of single agent apprenticeship learning algorithm in [1] to remove the dependence on quadratic programming (QP) solver in our algorithm. The result is shown in Fig.10. Note that, similar to the algorithm in [1], the performance guarantees of our algorithm rely on (approximately) matching the average empirical return. When agents depart from their start point, their policies are moving up and moving down, respectively. If they all move successfully, then the joint state is changed to (A in square 2, B in square 1) and the game is over. In this case, even though we may never recover the expert's reward function, our algorithm provides an explanation of the expert trajectories. The policy output by our algorithm can attain performance close to that of the expert, where here performance is measured with respect to the expert's unknown reward function.

7 CASE STUDY

Now, we infer passengers' preferences from their interactive choices in real-world data. Given thousands of transit passengers, only those whose trips overlapped spatially and temporally have influence to each other's decisions as in a Markov Game. Hence, passengers are

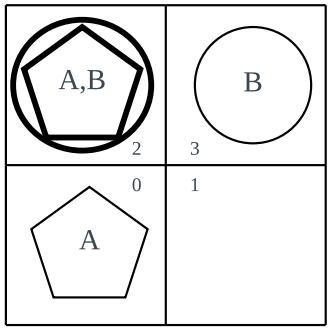

Figure 6: 2×2 grid game

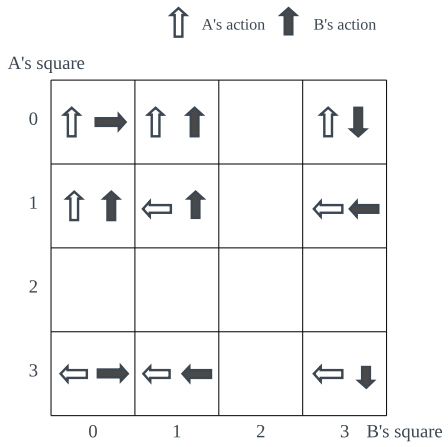

Figure 7: A Nash equilibrium joint policy

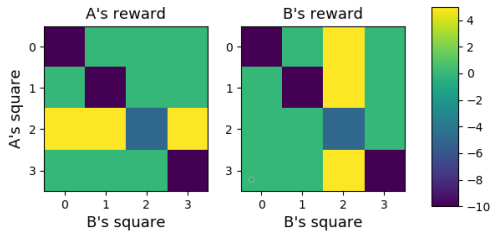

Figure 8: True reward functions for both agents

divided into small groups. In this section we use two such groups as examples to demonstrate the results. Fig.12 (depicted by LTMap [14]) shows a real-world case where passengers are traveling from Cuizhu to Grand Theatre. Travelers usually choose three transport modes. The first one is by subway. As is shown by the red line in Fig.12, passengers can take subway line #3 at Cuizhu and then transfer to subway line #1 at Laojie (denoted by the blue point in the figure). There are also two bus routes that directly arrive Grand Theatre from Cuizhu: bus route #29 (depicted by the blue

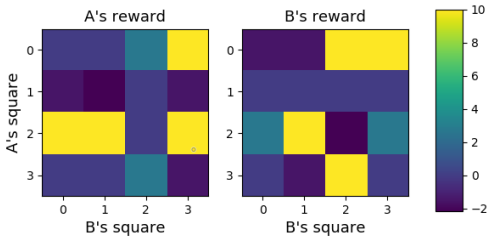

Figure 9: Predicted reward functions for both agents(from policy)

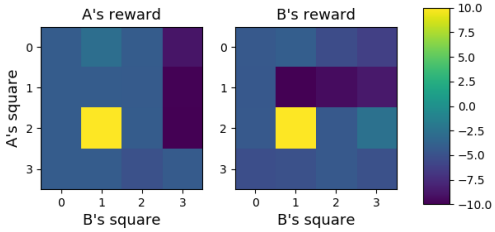

Figure 10: Predicted reward functions for both agents(from sampled trajectories)

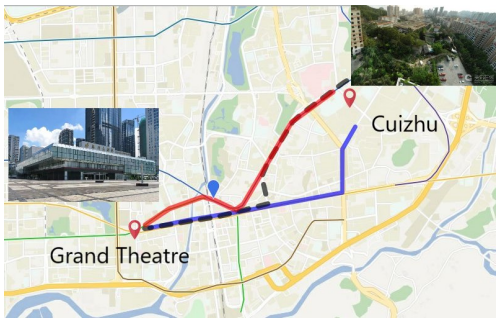

Figure 11: Sampled passengers' trajectories from Cuizhu to Grand Theatre

real line) and #3 (depicted by the black dotted line). Bus route #29 is cheap but slow, while bus route #3 is fast but a little costly. We sampled 5 passengers' trajectories from Cuizhu to Grand Theatre with the same departure time. One of them choose to take the subway, while two passengers choose to take bus route #29 with two passengers choosing bus route #3. We use agent ID 1-5 to denote these passengers.

Taking the advantage of Algorithm 1 proposed, we extracted these passengers preferences. Table 1 shows the result, where the value greater than zero represents the extend the passenger's preference degree while the value less than zero represents the degree of dislike. We can find that agent 1, who takes subway line #3 at first and then transfers to subway line #1, with very high reward weight in number of choices, prefers those transports which are more flexible. Besides, this passenger gains the lowest reward when the degree of crowding is high, which means that he/she will probably

Table 1: Passengers' preferences of five feature in case 1 (Cuizhu - Grand Theatre)

Passenger ID (Transit mode)	MC	RTT	NoT	NoC	DoC
1 (subway)	7.432e-02	6.224e-03	**7.174e-02**	**5.022e-02**	**-9.756e-01**
2 (bus route #29)	**-9.412e-02**	5.405e-02	-2.498e-16	0.000e+00	-3.040e-02
3 (bus route #29)	**-1.465e-01**	2.741e-01	9.256e-03	6.479e-03	-2.166e-01
4 (bus route #3)	2.473e-01	**-3.548e-01**	-6.477e-03	-4.534e-03	2.328e-01
5 (bus route #3)	-5.694e-02	**-9.299e-02**	-9.388e-02	-6.571e-02	1.135e-01

Figure 12: Sampled passengers' trajectories from Shenzhenbei Station to Shangmeilin

choose another transit mode if a certain bus route is crowded. Consider those travelers who take bus for a direct arrival, their weights of number of transfers and number of choices are low, denoting their preference of direct arrival. For those travelers who travel by bus route #29, a cheap but slow bus route, they care about the monetary cost a lot and can still gain a high reward if the trip takes a long time. For those who choose bus route #3 which is expensive but fast, on the contrary, they show little concerned of monetary cost, but they are likely to choose a transit mode which takes a short time. The results show that our algorithm is convergent in dealing with real data and that the algorithm can provide an explanation of passengers' trajectories. Especially, the interaction of passengers (the feature DoC in the table) is also extracted by our algorithm, which the algorithms of single agent IRL can not recover.

Fig.12 represents another real-world case of a group of travelers' MG. By sampling from the dataset of the public transport data from Shenzhen China, we obtained the choices of a group of 6 passengers traveling from Shenzhenbei Station to Shangmeilin at 8:30 a.m.: 3 of them chose the subway line #4 (the left, black line in the figure) for a direct arrival, while 2 passengers chose the bus route #324 (the right, blue line) and they also arrived Shangmeilin directly; only 1 traveler choses the bus route #M401 at first at transfer to subway line #4 at Minle (represented by the middle, red dotted line in Fig.12) since #M401 did not offer a direct arrival. Their trajectories are shown by lines in Fig.12. Using the algorithm proposed in Section 5.2, we extracted their preferences shown in table 2. We can see that the algorithm also recover passengers' interaction successfully. Passengers 3-5, taking the same subway together at the very beginning, can gain a high reward even the public transit is crowded, unsurprisingly. And passenger 6 who dislikes those crowded transit modes, prefers to choose flexible transits. Results of these two examples suggests that our algorithm can extract passengers' preferences on transit mode choice explained by features and can explain the interactions between passengers.

8 RELATED WORK

In this section, we summarize the literature works in the areas related to our study: urban computing, user choice modeling, mobility modeling and Human–Computer Interaction (HCI).

8.1 Urban Computing

Urban computing contains processes of acquiring, integrating, as well as analysing big and heterogeneous data generated in urban spaces to deal with the major issues that cities face [31]. Urban computing is an interdisciplinary field that integrates computer science with traditional areas (e.g., transportation, economics, road networks, sociology, civil engineering, and ecology) in the context of urban space. In [17], the authors propose a CityLines system which routes urban trips among spokes by a few direct paths or hubs, making fast and cheap trip possible. The authors in [3] propose a novel data-driven approach to the development of bike lane plans with the large-scale real-world bike trajectories. In [26], [30], the authors propose effective and real-time urban event detection approaches. The authors in [21] make the first attempt to characterize the dynamics of taxi drivers' preferences over time. In [29], authors investigate how to quantify human preferences in urban transit planning. However, none of those works have explicitly studied the "urban interactive human factors", i.e., when travelers make decisions together, how will a person's policy influenced by others. Our work is the first study investigating passengers' interaction on public transits.

8.2 User Choice Modeling

User choice modeling, which investigates how travelers make decisions, has been extensively studied. For instance, In [23], authors focus on the choice mechanism of park-and-ride users in choosing PNR lots, using two different approaches, random utility maximization and random regret minimization. In [33], the authors propose a probabilistic approach to extract drivers' route preferences when the collected data is inherently noisy and imperfect. What makes our work different from these works is that we employ data-driven approaches to model a unique real-world decision-making case.

Table 2: Passengers' preferences of five feature in case 2 (Shenzhenbei Station - Shangmeilin)

Passenger ID (Transit mode)	MC	RTT	NoT	NoC	DoC
1 (bus route #324)	**2.878e-01**	**-5.216e-01**	-1.168e-01	-3.85e-02	1.877e-01
2 (bus route #324)	**4.401e-01**	**-4.567e-01**	-3.124e-01	-1.030e-01	3.217e-01
3 (subway line #4)	-2.420e-03	6.798e-01	-3.752e-01	-1.238e-01	**6.575e-01**
4 (subway line #4)	4.738e-02	5.360e-01	-1.833e-01	-6.050474e-02	**6.525e-01**
5 (subway line #4)	-3.470e-03	2.292e-01	-4.167e-01	-1.375e-01	**2.041e-01**
6 (bus#M401 - subway #4)	-3.065e-01	8.703e-02	**2.489e-01**	**8.216e-02**	**-7.095e-01**

8.3 Mobility Modeling

Mobility model plays an important role in examining different issues involved in a cellular system such as user location updating and the like. In general, the mobility modeling should include changes in both the direction and speed of the mobile [34]. Some papers focus on modeling and tracing the path of a mobile. In [4], the authors propose a hierarchal hidden semi-Markov model to address the problem of modeling movement tracks of mobile objects. This technique can model the movement both in stay-points and in the paths connecting them. In [2], authors present a novel approach for modeling human routine behavior from behavior logs that explicitly models the causal relationship between the contexts and actions the passengers perform in those contexts. It is shown that routine models extracted using this approach can help researchers to identify routines and routine variations without having to manually search for those patterns in raw data. Our work makes the first effort to inferring passengers' interactive choices on public transits, so we propose a novel Markov Game model to model passengers' trajectories.

8.4 Human–Computer Interaction

In [12], authors have done substantial work in finding that transit use is often shaped around the transparency that systems have in their predictions. Therefore, they determine the general design requirements for visualizing uncertainty on mobile applications, as well as the domain specific design requirements for visualizing uncertainty in transit arrival times. And they propose a mobile interface to visualize the uncertainty in real-time traffic prediction in a way that supports users' goals. Our work provides a general framework on evaluating how various factors affect passengers' decisions, i.e., the extracted preference vector can be viewed as weights passengers consider for different features. Hence, our framework can provide a way for counter-factual reasoning and supplementary evidence for the HCI community, where they use case studies and surveys to analyze the impacts of various factors to human decisions.

9 CONCLUSION AND FUTURE WORK

In this paper, we make the first attempt to model passengers' traveling by public transports as MGs. We then propose two novel algorithms to extract agents' reward functions in MA-IRL problems base on deterministic equilibrium policies. Then we use our MA-AL algorithm on real-world data from Shenzhen, China to infer passengers' interactive preferences by identifying passengers' trajectories as a Nash equilibrium joint policy. The results show that our algorithms have the ability to infer passengers' interactive transit mode choices explained by a set of features.

An exciting avenue for future work is to use the passengers' preferences extracted from MA-AL to evaluate passengers' future transport choices, estimating the ridership before deploying a new subway line or a new bus route. Hence, the downstream applications of our work include urban transportation planning, such as subway lines and bus routes planning. Our future work include smart urban transit planning by employing the passenger preferences extracted from our MA-AL (since the passengers preferences can enable accurate ridership prediction for transit deployment plans); and generalizing MA-AL framework to non-linear preference function, using deep neural networks.

ACKNOWLEDGMENTS

Yanhua Li was supported in part by NSF grants CNS-1657350 and CMMI-1831140, and a research grant from DiDi Chuxing Inc. Hui Lu and Zhihong Tian were supported by the Guangdong Province Key Area R&D Program of China (2019B010137004), National Natural Science Foundation of China (U1636215, 61972108, 61871140), the National Key research and Development Plan (2018YFB0803504), and Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019).

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 1.
[2] Nikola Banovic, Tofi Buzali, Fanny Chevalier, Jennifer Mankoff, and Anind K Dey. 2016. Modeling and understanding human routine behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 248–260.
[3] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. 2017. Planning bike lanes based on sharing-bikes' trajectories. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1377–1386.
[4] Mitra Baratchi, Nirvana Meratnia, Paul JM Havinga, Andrew K Skidmore, and Bert AKG Toxopeus. 2014. A hierarchical hidden semi-Markov model for modeling mobility data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 401–412.
[5] Dimitri P Bertsekas and John N Tsitsiklis. 1996. *Neuro-dynamic programming*. Vol. 5. Athena Scientific Belmont, MA.
[6] Adrian A Canutescu and Roland L Dunbrack Jr. 2003. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein science* 12, 5 (2003), 963–972.
[7] Jerzy Filar and Koos Vrieze. 1997. Competitive Markov Decision Processes-Theory, Algorithms, and Applications. (1997).
[8] Google GeoCoding. 2016. *Road map data*. Google. https://developers.google.com/maps/documentation/geocoding/
[9] Amy Greenwald, Keith Hall, and Roberto Serrano. 2003. Correlated Q-learning. In *ICML*, Vol. 3. 242–249.
[10] Peter Henry, Christian Vollmer, Brian Ferris, and Dieter Fox. 2010. Learning to navigate through crowded environments. In *2010 IEEE International Conference*

on Robotics and Automation. IEEE, 981–986.
[11] Junling Hu, Michael P Wellman, et al. [n.d.]. Multiagent reinforcement learning: theoretical framework and an algorithm. Citeseer.
[12] Matthew Kay, Tara Kola, Jessica R Hullman, and Sean A Munson. 2016. When (ish) is my bus?: User-centered visualizations of uncertainty in everyday, mobile predictive systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5092–5103.
[13] Volodymyr Kuleshov and Okke Schrijvers. 2015. Inverse game theory: Learning utilities in succinct games. In *International Conference on Web and Internet Economics*. Springer, 413–427.
[14] LDMap. 2019. *Road map*. DIGITALAND. http://www.ldmap.net/index.html
[15] Yanhua Li, Jun Luo, Chi-Yin Chow, Kam-Lam Chan, Ye Ding, and Fan Zhang. 2015. Growing the charging station network for electric vehicles with trajectory data analytics. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 1376–1387.
[16] Yanhua Li, Moritz Steiner, Jie Bao, Limin Wang, and Ting Zhu. 2014. Region sampling and estimation of geosocial data with dynamic range calibration. In *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 1096–1107.
[17] Guanxiong Liu, Yanhua Li, Zhi-Li Zhang, Jun Luo, and Fan Zhang. 2017. Citylines: Hybrid hub-and-spoke urban transit system. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 51.
[18] John Nash. 1951. Non-cooperative games. *Annals of mathematics* (1951), 286–295.
[19] Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning.. In *Icml*, Vol. 1. 2.
[20] OpenStreetMap. 2016. *Road map data*. http://www.openstreetmap.org/
[21] Menghai Pan, Yanhua Li, Xun Zhou, Zhenming Liu, Rui Song, Hui Lu, and Jun Luo. 2019. Dissecting the Learning Curve of Taxi Drivers: A Data-Driven Approach. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 783–791.
[22] I.O.York R.J.Balcombe and D.C.Webster. 2003. *Factors Influencing Trip Mode Choice*. Technical Report. Transport Research Labortory.
[23] Bibhuti Sharma, Mark Hickman, and Neema Nassir. 2019. Park-and-ride lot choice model using random utility maximization and random regret minimization. *Transportation* 46, 1 (2019), 217–232.
[24] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*. 7461–7472.
[25] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*.
[26] Amin Vahedian, Xun Zhou, Ling Tong, Yanhua Li, and Jun Luo. 2017. Forecasting gathering events through continuous destination prediction on big trajectory data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 34.
[27] Senzhang Wang, Lifang He, Leon Stenneth, Philip S Yu, and Zhoujun Li. 2015. Citywide traffic congestion estimation with social media. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 34.
[28] Kevin Waugh, Brian D Ziebart, and J Andrew Bagnell. 2013. Computational rationalization: The inverse equilibrium problem. *arXiv preprint arXiv:1308.3506* (2013).
[29] Guojun Wu, Yanhua Li, Jie Bao, Yu Zheng, Jieping Ye, and Jun Luo. 2018. Human-Centric Urban Transit Evaluation and Planning. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 547–556.
[30] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance Kaplan, Shaowen Wang, and Jiawei Han. 2016. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 513–522.
[31] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 38.
[32] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. 2010. Modeling interaction via the principle of maximum causal entropy. (2010).
[33] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. (2008).
[34] Mahmood M. Zonoozi and Prem Dassanayake. 1997. User mobility modeling and characterization of mobility patterns. *IEEE Journal on selected areas in communications* 15, 7 (1997), 1239–1252.