Is Reinforcement Learning the Choice of Human Learners? A Case Study of Taxi Drivers

Menghai Pan, Weixiao Huang
mpan,whuang2@wpi.edu
Worcester Polytechnic Institute

Yanhua Li
yli15@wpi.edu
Worcester Polytechnic Institute

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

Zhenming Liu
zliu@cs.wm.edu
College of William & Mary

Jie Bao, Yu Zheng
baojie@jd.com,msyuzheng@outlook.com
JD Finance

Jun Luo
jluo1@lenovo.com
Lenovo Group Limited

ABSTRACT

Learning to make optimal decisions is a common yet complicated task. While computer agents can learn to make decisions by running reinforcement learning (RL), it remains unclear how human beings learn. In this paper, we perform the first data-driven case study on taxi drivers to validate whether humans mimic RL to learn. We categorize drivers into three groups based on their performance trends and analyze the correlations between human drivers and agents trained using RL. We discover that drivers that become more efficient at earning over time exhibit similar learning patterns to those of agents, whereas drivers that become less efficient tend to do the opposite. Our study (1) provides evidence that some human drivers do adapt RL when learning, (2) enhances the deep understanding of taxi drivers' learning strategies, (3) offers a guideline for taxi drivers to improve their earnings, and (4) develops a generic analytical framework to study and validate human learning strategies.

KEYWORDS

human learning strategy, urban computing, reinforcement learning

ACM Reference Format:
Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, Zhenming Liu, Jie Bao, Yu Zheng, and Jun Luo. 2020. Is Reinforcement Learning the Choice of Human Learners? A Case Study of Taxi Drivers. In *Proceedings of Seattle '20: 28th ACM SIGSPATIAL International Conference on Advances in Gepgraphic Information Systems (SIGSPATIAL '20).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Learning to make decisions is ubiquitous for human beings. For example, a Go player learns to imitate other players to devise better game strategies. A physician learns to determine doses of drugs through extensive case studies and sometimes ad-hoc experimentation. A professional driver learns to cruise through repeated practice to effectively find the next passenger. While a learning process is often complex, recent advances in machine learning have enabled computer agents to automate some learning tasks. For example, reinforcement learning (RL) is used to train AlphaGo [44] to beat the human champion and build systems to recommend medical treatments [50]. Optimizing taxi operation strategies has also been extensively studied in the literature [31, 37, 41, 59, 62]. Many recent solutions also rely on RL techniques.

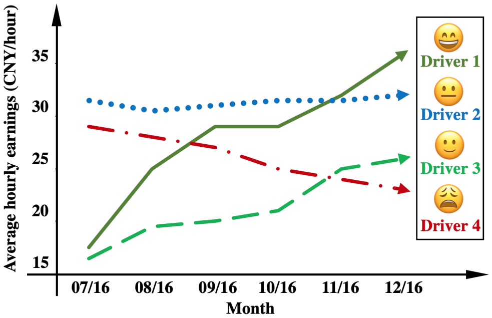


Figure 1: Diverse patterns of drivers' per-hour income dynamics in Shenzhen, China.

While progress was made to design RL algorithms for computer agents to learn, it remains unclear how the human counterpart learns. Do human learning processes exhibit similar patterns to the one driven by RL algorithms, or they deviate from any known learning strategies? Answering this problem is important for three reasons: *(i)* many decision-making problems remain challenging for machines and still require "human learning", so it becomes important to distill decision strategies from humans; *(ii)* effective humans learning strategies can be used to train beginners such as new Go players and new taxi drivers; and *(iii)* it also advances cognitive and social science research by taking an algorithmic lens at human learners' behaviors.

This paper examines how traditional taxi drivers learn to cruise for seeking their next passengers. Here, traditional drivers refer to those that do not rely on mobile-based platforms such as Uber, Lyft, or DiDi. These drivers represent a significant portion of personnel in taxi service, despite recent growth of online platforms. Prior studies [37] on this problem assumed drivers are rational and use inverse reinforcement learning to characterize drivers' behaviors. Although these works offer useful insights, not all drivers are rational. Some drivers learn faster than others. Some drivers' performance even deteriorates over time. For example, Fig. 1 shows

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the dynamics of per-hour incomes from four typical taxi drivers in 2016 in Shenzhen, China. Driver 1 (dark green line) started at a relatively low-income level, but then rapidly doubled the income level by the end of the year. Driver 2 (light green line) started at a similar (low) income level as Driver 1, but had a much slower increasing trend. Driver 3 (blue line) had a stable income level over time. Moreover, the income level of Driver 4 (red line) went down roughly by 30% in six months.

This example suggests that different drivers use different strategies to learn. Thus, our work focuses on investigating i) *what learning strategies they are following, especially for those "quick learners"?;* ii) *how do these strategies compare to what a computer agent would follow in reinforcement learning?*

Specifically, we investigate and validate human learning strategies through a data-driven case study on taxi drivers. To the best of our knowledge, this is the first attempt of its kind in the context of taxi operations. Specifically, we extract trips of taxi drivers from a large-scale dataset spanning 6 months with over 17,000 taxis. We categorize drivers into different groups based on their hourly earning dynamics. For each group of drivers, we build estimation procedures to construct the time series of a driver's policy and advantage functions and examine whether their patterns are consistent with those of an agent in a RL algorithm. In addition, we validate under what scenarios the drivers are following the paradigm of RL, if not always.

Our major finding is that a taxi driver's improvement in earning efficiency is positively correlated with how well he/she follows the process of RL algorithm. In addition, human drivers usually do not completely follow RL when learning. They tend to follow RL first for those scenarios (e.g., certain urban areas) that lead to higher earning improvement. *Our contributions* are summarized as follows:

(1) We propose a three-stage analytical framework to rigorously validate whether human agents (e.g., taxi drivers) follow RL paradigms to improve their earning efficiencies.

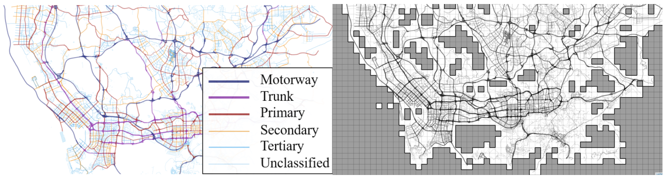
(2) It is evident from the analytical results on a large-scale taxi trajectory dataset that successful drivers are likely those who follow the RL paradigm better. Moreover, they tend to follow RL first for those scenarios (e.g., certain urban areas) that lead to higher earning improvement. *We made our code and unique dataset publicly available to contribute to the research community [2].*

2 OVERVIEW

In this section we present our problem statement, followed by an introduction to the dataset and our solution framework.

Problem Definition: Given real trajectory data of taxi drivers $\tilde{\mathcal{T}}$ in a sequence of time intervals $T_0, T_1, ..., T_n$, we aim to validate or reject the following hypotheses: (1) Drivers that are successful in learning passenger-seeking experiences (i.e., with increasing earning efficiency), employ learning strategies that are closer to reinforcement learning (RL) paradigms; (2) The RL paradigm is followed by human drivers only at certain scenarios (e.g., locations, times) rather than all circumstances. We also aim to identify what these scenarios are.

Dataset: We use two data sources: *(i)* taxi trajectory data and *(ii)* road map data, both collected in Shenzhen, China in 2016.



(a) Shenzhen road map **(b) Map gridding**

Figure 2: Shenzhen map data

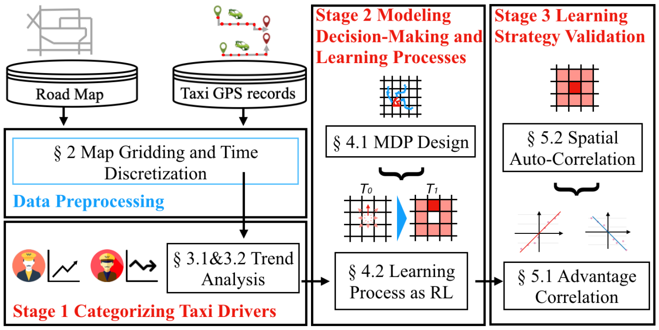

Road Map
Taxi GPS records
§ 2 Map Gridding and Time Discretization
Data Preprocessing
§ 3.1&3.2 Trend Analysis
Stage 1 Categorizing Taxi Drivers
Stage 2 Modeling Decision-Making and Learning Processes
§ 4.1 MDP Design
T_0
T_1
§ 4.2 Learning Process as RL
Stage 3 Learning Strategy Validation
§ 5.2 Spatial Auto-Correlation
§ 5.1 Advantage Correlation

Figure 3: Solution framework

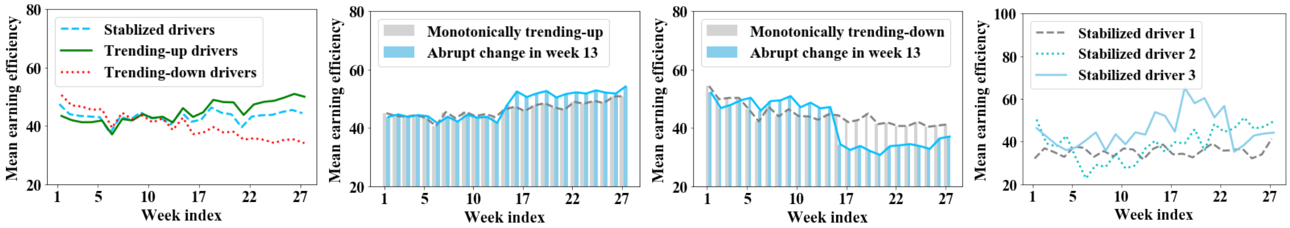
The **taxi trajectory data** contains GPS records collected from taxis in Shenzhen, China between July 1st and December 31st in 2016. There are in total 17, 877 taxis equipped with GPS sets. Each GPS set generates a GPS point every 40 seconds on average. A total of 51, 485, 760 GPS records are collected on a daily basis. Each record contains five fields, including taxi ID, time stamp, passenger indicator, latitude, and longitude. The passenger indicator field is a binary value, indicating if a passenger is aboard or not.

The **Road map data** was collected from [1]. It covers the rectangular area between 22.44° to 22.87° in latitude and 113.75° to 114.63° in longitude. This area includes 21, 000 road segments with six levels, including motorway, trunk way, primary road, secondary road, tertiary, and unclassified, as shown in Fig. 2a.

Data Preprocessing: We preprocess the datasets by map gridding and time discretization.

(1) Map gridding. The urban road network forms a continuous space. We use the gridding-based method to simply partition the road map into equally sized grids [28, 29]. This method is easy to implement and make adjustment. It allows us to adjust the size of the grids, and examine the impact of the grid size. We let s be the side-length of each cell. Cells adjacent to each other are considered reachable if there is at least one road across their boundary. Fig. 2b visualizes of our gridding results with side-length of $s = 0.01°$ in latitude and longitude. By removing grid cells in those unreachable regions in the city (e.g., in the center of a part), we have a total of $n = 1, 018$ valid cells (highlighted in light colors in Fig. 2b) covered by the road network.

(2) Time Discretization. We divide each day (24 hours) into three time intervals, i.e., 00:00 − 06:00, 06:00 − 16:00, and 16:00 − 24:00, based on the common schedules of taxi drivers. In Shenzhen, each taxi is usually operated by two drivers. One driver operates in day time and the other operates at nights. Thus, taxi trajectories in different time intervals are considered from different drivers. Two



(a) Mean earning efficiency of each group

(b) Trending-up drivers

(c) Trending-down drivers

(d) Stabilized drivers

Figure 4: Different trends of earning efficiencies in the three driver groups

drivers usually switch at around 6AM and 4PM everyday. Finally, because there are exceedingly small numbers of taxi trips between mid-night and early morning, we focus on only two time intervals, i.e., 06:00 - 16:00 and 16:00 - 24:00.

Solution Framework: Our proposed solution framework is outlined in Fig. 3, which takes two sources of urban data as inputs and contains three analytical stages: (1) categorizing taxi drivers in section 3, (2) modeling decision-making and learning process in section 4, (3) learning strategy validation in section 5.

3 STAGE I: CATEGORIZING TAXI DRIVERS

This section introduces the definition of taxi drivers' earning efficiencies (Section 3.1), the earning efficiency dynamics of taxi drivers (Section 3.2), and classification of taxi drivers based on the trends of their earning efficiencies (Section 3.3).

3.1 Quantifying Taxi Drivers' Earning Efficiencies

To quantify the earning efficiencies of taxi drivers, we need to address two issues: *1. Effective working hours. 2. Changes in earning efficiencies.* Drivers' earning efficiencies evolve over time. Thus, we re-estimate drivers' earning efficiencies every week.

Let r_e^i be the earning efficiency of driver e in week i $(1 \leq i \leq 27)$. We let

$$r_e^i = \frac{E_e^i}{t_e^i}, \tag{1}$$

where E_e^i is his/her total income in week i and t_e^i is the total working hours. Here, the total working hours are the time when the driver is seeking for passengers or serving passengers. We eliminate the time when the driver takes a break (the taxi stays still for 30 minutes or more).

3.2 Earning Efficiency Trend Analysis

We aim to detect the following patterns in drivers' earning efficiencies changes:

- *Monotonic increase/decrease.* The increase or decrease occurs constantly over the entire time series.
- *Abrupt increase/decrease.* At a certain time point, an abrupt increase or decrease occurs, differing the statistics of time series before and after that significantly.

When the efficiency of a driver does not exhibit any of the above changes, we define the driver as a *stabilized* driver. Fig. 4(b)-(d) show examples of different learner groups. Next, we devise two statistical tools to detect the aforementioned patterns.

Mann-Kendall (MK) Trend Test [15] is a hypothesis test method for monotonic trend in time series data, which indicates whether a trend exists and whether the trend is positive or negative. The null hypothesis H_0 is **no monotonic trend**, while the alternative hypothesis H_1 is **monotonic trend is present**.

The statistic of Mann-Kendall test can be calculated as follows,

$$Z_{MK} = \begin{cases} \frac{S-1}{\sqrt{VAR(S)}} & if\ S > 0, \\ 0 & if\ S = 0, \\ \frac{S+1}{\sqrt{VAR(S)}} & if\ S < 0, \end{cases} \tag{2}$$

$$S = \sum_{k=1}^{n-1} \sum_{j=k+1}^{n} \operatorname{sgn}(r_e^j - r_e^k), \tag{3}$$

$$\operatorname{sgn}(r_e^j - r_e^k) = \begin{cases} 1 & if\ r_e^j - r_e^k > 0, \\ 0 & if\ r_e^j - r_e^k = 0, \\ -1 & if\ r_e^j - r_e^k < 0, \end{cases} \tag{4}$$

$$VAR(S) = \frac{1}{18}[n(n-1)(2n+5)]. \tag{5}$$

Given a confidence α, the null hypothesis is rejected if $|Z_{MK}| > Z_{1-\alpha}$, where $Z_{1-\alpha}$ is the $(100(1-\alpha))^{th}$ percentile of the standard normal distribution.

Pettitt's Test [22] is to detect change points in time series data. A change point in a time series $r_e^1, r_e^2, r_e^3, ..., r_e^t, ..., r_e^n$ refers to a time index t such that $\{r_e^{t_1}\}_{t_1 \leq t}$ and $\{r_e^{t_2}\}_{t_2 > t}$ follow two distributions [16]. The null hypothesis H_0 is **no abrupt change points exist**, while the alternative hypothesis H_1 is **an abrupt change point exists**.

Pettitt's test uses a non-parametric test statistics U_t defined as

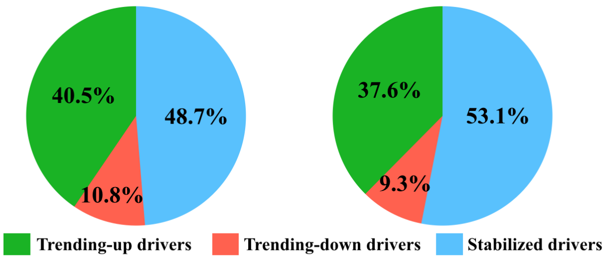
$$U_t = \sum_{i=1}^{t} \sum_{j=t+1}^{n} \operatorname{sgn}(r_e^i - r_e^j). \tag{6}$$

Then we can calculate:

$$K = \max_{1 \leq t \leq n} U_t. \tag{7}$$

The change-point of the series is located at time K, provided that the statistic is significant. The significance probability of K is approximated for $p \leq 0.5$ with:

$$p \approx 2 \exp \frac{-6K^2}{n^3 + n^2}. \tag{8}$$



(a) 6am-4pm (b) 4pm-12am

Figure 5: Driver groups

3.3 Results on Trend Analysis

We next describe our result. Our dataset contains 2,403 taxis in the 6am to 4pm interval and 2,790 taxis in the 4pm to 12am interval. We categorize taxi drivers into three groups: **(1) Trending-up:** if at least one of the tests (MK and Pettitt) show significant increasing trend, **(2) Trending-down:** if at least one of the tests show significant decreasing trend, and **(3) Stabilized** if none of the tests is significant. The two tests do not produce any inconsistent conclusions among drivers we examine (i.e., one test shows it trends up whereas the other shows it trends down).

Fig. 5 presents the results. Note Week #14 and #15 are excluded from the dataset because they have much smaller trip numbers due to the national holiday. This is to avoid biased results.

We can see that around half of the drivers are stabilized drivers, and the number of trending-up drivers is larger than the number of trending-down drivers in both intervals. Fig. 4a shows the average earning efficiencies for each group of drivers over 25 weeks. The trends exhibit here are consistent with the test results.

4 STAGE II: MODELLING DECISION-MAKING AND LEARNING PROCESSES

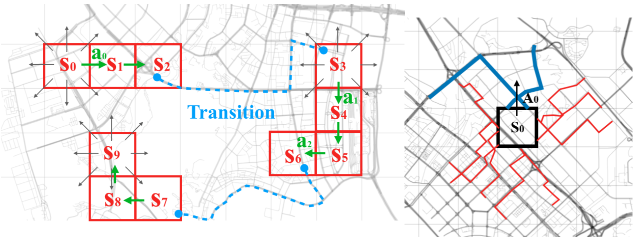
We next model the drivers' behaviors. We need to model: *(i) how drivers make decisions* (i.e., how they look for and serve passengers). This is modeled by a Markov Decision Process (Sec 4.1). *(ii) how drivers learn to make decisions* (i.e., how they use their past experience to update their decision policies over time). Based on our hypothesis, we use reinforcement learning (RL) to model this process (Sec 4.2).

4.1 Decision-Making Process as an MDP

A taxi driver needs to determine the travel direction when the taxi is idle and this decision impacts his/her chance to find a new passenger. We model this decision-making process as a Markov Decision Process (MDP) [4].

Review of MDP. An MDP is represented as a 5-tuple $\langle S, A, T, \gamma, \mu_0, R\rangle$.

- S is a finite set of states;
- A is a finite set of actions;
- T is the probabilistic transition function with $T(s'|s,a)$ as the probability of arriving at state s' by executing action a at state s;
- $\gamma \in (0,1]$ is the discount factor[1];



(a) MDP of taxi driver's decision-making process (b) $Q(S_0, A_0)$, $V(S_0)$

Figure 6: Illustrations of MDP and RL concepts

- $\mu_0 : S \to [0,1]$ is the initial state distribution;
- $R : S \times A \to \mathbb{R}$ is the reward function.

A randomized, memoryless policy is a function that specifies a probability distribution on the action to be executed in each state, defined as $\pi : S{\times}A \to [0,1]$. We use $\tau = [(s_0,a_0),(s_1,a_1),\ldots,(s_L,a_L)]$ to denote a trajectory generated by MDP. Here L is the length of trajectory.

Applying MDP to model drivers. We model the decision-making process of taxi drivers with MDP as follow:

- State: a spatial region, specified by a geographical grid cell, created with map gridding in data preprocessing phase;
- Action: traveling from the current cell to one of the eight neighboring cells.

Fig. 6a shows an example of taxi trajectory as an MDP: a driver starts in state s_0 with the taxi idle, and takes the action a_0 to travel to the neighboring cell S_1 on the right. After two decisions, the driver traverses S_1 and reaches state S_2, where a passenger is found at S_2. Then, a passenger trip corresponds to a transition in the MDP from starting state S_2 to ending state S_3. Each decision made at a certain state would lead to a reward as the expected monetary income of finding and serving a passenger. The policy π employed by a driver is a probability distribution of choosing each action at each state.

4.2 Learning Process as Reinforcement Learning

Hypothesis. When one starts working as a taxi driver, he/she may not have knowledge about where to find the next passenger, and may choose a simple initial policy π_0. Over time, the driver learns from his/her experience and update the policy to π_1 with a goal to increase his/her income. The driver repeats this process so his/her policy evolves continuously (see also [41, 59, 62]).

Types of reinforcement learning. Reinforcement learning (RL) algorithms can be classified into three major categories including value-based RL [47], policy-based RL [48], Actor-Critic based approach [25]. We briefly outline the key ideas of the three types of RL algorithms below. A key similarity of all these algorithms is that they optimize the policy functions by "taking the gradient" with respect to the advantage function, which is defined as the additional reward gained from the current policy comparing to the one in the previous iteration.

• *Value-based RL* [47] does not learn the optimal policy directly. It learns the so-called Q value (or V value) instead, which is defined on each state-action pair (s,a), namely, $Q(s,a)$ (or on each state s, namely, $V(s)$). Specifically, $Q(s,a)$ refers to the expected future

[1] Without loss of generality, we assume $\gamma = 1$ in this study, and it is straightforward to generalize our results to $\gamma \neq 1$.

Table 1: Typical methods of RL

	Typical Method	Update function
Value-based Methods	Q-learning	$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)]$ [47]
	SARSA	$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) + \gamma Q(s', \pi_Q(s')) - Q(s,a)]$ [47]
Actor-Critic Methods	Actor-Critic	$\nabla\overline{R_\theta} \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(Q^{\pi_\theta}(s_t^n, a_t^n) - V^{\pi_\theta}(s_t^n))\nabla \log p_\theta(a_t^n \mid s_t^n)$ [25]
	Advantage Actor-Critic(A2C)	$\nabla\overline{R_\theta} \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(r_t^n + V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n))\nabla \log p_\theta(a_t^n \mid s_t^n)$ [25]
Policy-based Methods	Policy gradient	$\theta \leftarrow \theta + \alpha\nabla\overline{R_\theta}, \nabla\overline{R_\theta} \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(\sum_{t'=t}^{T_n}\gamma^{t'-t}r_{t'}^n - b)\nabla \log p_\theta(a_t^n \mid s_t^n)$ [48]

reward, after taking an action a at a state s, while $V(s)$ refers to the expected reward after leaving a state s. Once Q-functions are well learned, the optimal policy π^* can be recovered from the optimal value function of each state-action pair (e.g., $Q(s,a)$). The Q-learning [47] and State-Action-Reward-State-Action (SARSA) methods [47] are the state-of-the-art value-based RL algorithms.

- *Policy-based RL* [48] learns an optimal policy directly. Usually, policy π is represented by a (deep) neural network with parameter set θ. A well known policy-based method is policy gradient [48]. The objective of policy gradient is to maximize the expected future reward over trajectories:

$$\max_\theta \overline{R_\theta} = \max_\theta\{\mathbf{E}(R_\theta)\} = \max_\theta\{\sum_\tau R(\tau)p_\theta(\tau)\}. \tag{9}$$

Where $R(\tau)$ is the accumulated reward in trajectory τ and $p_\theta(\tau)$ denotes the probability of generating trajectory τ under the policy with parameter θ. Then, we can apply gradient ascent to find the optimal θ. The gradient of the objective function with respect to θ is:

$$\nabla\overline{R_\theta} \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(\sum_{t'=t}^{T_n}\gamma^{t'-t}r_{t'}^n - b)\nabla \log p_\theta(a_t^n|s_t^n), \tag{10}$$

where N is the number of trajectories, T_n is the length of trajectory n, t and t' are the time steps. b is the baseline, i.e., average reward received.

- *Actor-Critic based RL* [25] combines both value-based and policy based methods, $\sum_{t'=t}^{T_n}\gamma^{t'-t}r_{t'}^n$ is evaluated using $Q^{\pi_\theta}(s_t^n, a_t^n)$, and we can use V^{π_θ} to be the baseline b. Moreover, $Q^{\pi_\theta}(s_t^n, a_t^n) - V^{\pi_\theta}$ is denoted by $A^\theta(s_t, a_t)$ which is called the *advantage function.* If the expected reward after taking a state-action pair is higher than the average expected reward after exiting the state, i.e., the advantage is positive, the agent will increase the probability of taking this action in this state. The advantage function is used to update the gradient, which in turn updates the parameter of the policy network.

Similarities of three RL paradigms. The gradient update functions of the three typical methods of RL are listed in Table 1. They all try to maximize the expected accumulated reward in each state or state-action pair, which is related to $Q(s,a)$ and $V(s)$. In other words, all these RL algorithms are equivalent, in a sense that a larger advantage of an state-action pair results in a increased probability of choosing such pair in the future policy.

Empirical estimates. Our main goal is to validate whether the real-world learning process of the drivers is consistent with the policy gradient method. Here, we describe how the key variables/functions are estimated through data.

- *Estimation of advantage functions.* Recall that the advantage function captures the additional reward gained from the change of one's policy. We estimate the advantage function value of each state-action pair for each driver. In time span T_0, the advantage of a driver in each state-action pair can be estimated by the empirical Q value and empirical V value. The empirical Q value is the average earning efficiency of the driver within a certain range of time after exiting each state via each action, whereas the empirical V value is the average earning efficiency of the driver with a certain range of time after exiting each state. The difference between Q value and V value is that V value characterizes the expected reward after leaving each state s, while Q value characterizes the expected reward after taking each state-action pair (s,a) As shown in Fig. 6b, $V(S_0)$ is calculated using all red trajectories and blue trajectories, which are the service trips exiting S_0, whereas $Q(S_0, A_0)$ is calculated using only the blue trajectories, which are the service trips exiting S_0 through action A_0.

$$A(s,a) = Q(s,a) - V(s). \tag{11}$$

- *Estimation of policy functions and their differences.* We also need to estimate the difference of policies between two consecutive time spans, T_0 and T_1. The empirical policy $\pi(s,a)$ of each state-action pair in each time span can be estimated via the visitation frequencies,

$$\pi(s,a) = \frac{D(s,a)}{D(s)}, \tag{12}$$

where $D(s,a)$ and $D(s)$ denote the visitation frequency of the state-action pair (s,a) and state (s) respectively. Validating if taxi drivers follow RL is equivalent to examine if there exists significant correlation between the difference of policy $\Delta\pi(s,a)$ and the advantage $A(s,a)$. Next section continues the discussion of the validation process.

5 STAGE III: LEARNING STRATEGY VALIDATION

This section describes our validation process. This consists of *(i)* identifying the correlation between the policy difference and the advantage, and *(ii)* correcting spatial bias of the empirical policy difference and the advantage by analyzing the spatial auto-correlation.

5.1 Advantage Correlation

To validate if there exists a correlation between the policy difference $\Delta\pi(s,a)$ and the advantage $A(s,a)$, a correlation coefficient should be used. A common one is Pearson's correlation coefficient[5], but it has the assumption of independent and identical distribution of

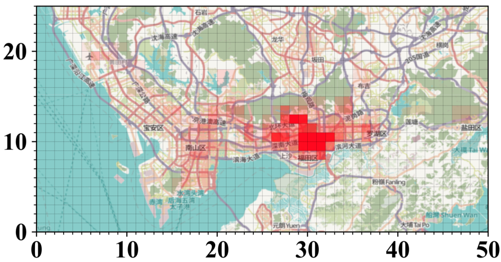


Figure 7: Heatmap of a driver's $D(s)$

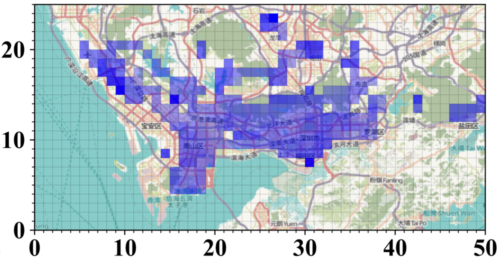
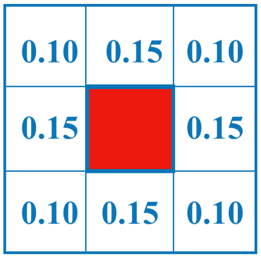


Figure 8: Heatmap of a driver's $V(s)$



0.10	0.15	0.10
0.15	🟥	0.15
0.10	0.15	0.10

Figure 9: Weight matrix

data. The Spearman's rank correlation coefficient [6] works for non-parametric data measuring a statistical relationship between two variables, which is more applicable in our ordinal data. Therefore, we employ **Spearman's rank correlation coefficient** in addition to the Pearson's correlation coefficient to evaluate the correlation coefficient and test its significance. The statistic of Spearman's rank correlation coefficient can be calculated by the formula below:

$$\rho = \frac{\sum_{i=1}^{n}(rank(A_i) - \overline{rank(A)})(rank(\Delta\pi_i) - \overline{rank(\Delta\pi)})}{\sqrt{\sum_{i=1}^{n}(rank(A_i) - \overline{rank(A)})^2 \sum_{i=1}^{n}(rank(\Delta\pi_i) - \overline{rank(\Delta\pi)})^2}}, \tag{13}$$

where A_i is the advantage of the $i-th$ sample, and $\Delta\pi_i$ is the policy difference of the $i-th$ sample. $rank$ denotes the ordinary rank of the corresponding value, and n is the sample size.

ρ ranges from -1 to 1, and the sign of ρ indicates the direction of the association between the advantage and the policy difference, e.g., if the sign is positive, the policy difference tends to decrease with the increase of the advantage.

We can also determine the significance of the ρ. We calculate the t value according to the formula below:

$$t = \rho\sqrt{\frac{n-2}{1-\rho^2}}. \tag{14}$$

Then we check the p value by calculating the t value according to the Student's t distribution.

5.2 Incorporating Spatial Auto-Correlation

Intuitively, nearby grids may cover the same urban functional zone in a city and share similar demand patterns. This can be observed from the real world data. Fig. 7 & 8 show the heatmaps of the $D(s)$ and $V(s)$ of a driver in July 2016, where we can observe that similar values are clustered. Therefore, it's reasonable to incorporate spatial auto-correlation when estimating $D(s)$ and $V(s)$.

(1) Quantifying spatial auto-correlation in $D(s)$ and $V(s)$.

Given a grid cell, we consider the eight neighboring grid cells as its spatial neighbors (i.e., the Queen neighborhood). A weight matrix is used to define the strength of correlation between pairs of locations, based on the inverse Manhattan distance between each pair of grid cells, i.e., the original weight w_{ij} between grid i and grid j ($i \neq j$) is:

$$w_{ij} = \begin{cases} \frac{1}{Manh\_dist(i,j)+1} & if\ neighbor(i,j) = True, \\ 0 & if\ neighbor(i,j) = False, \end{cases} \tag{15}$$

where $Manh\_dist(i,j)$ returns the Manhattan distance between grid i and grid j, and $neighbor(i,j)$ returns $True$ if grid i and j are neighboring and vise versa. Then the weights for each grid are normalized among its neighbors. Fig. 9 shows an example of the weights between the neighboring grids and the red grid.

Moran's I [13] is a measure of spatial auto-correlation. The statistic of Moran's I test can be calculated in Eq. 16

$$I = \frac{N}{W}\frac{\sum_i \sum_j w_{ij}(x_i - \overline{x})(x_j - \overline{x})}{\sum_i (x_i - \overline{x})^2}, \tag{16}$$

where x is the value of interest in each location, N is the number of spatial units, i, j are the indexes of two spatial locations, w_{ij} is the weight between location i and location j, W is the sum of all w_{ij}. Value of I ranges from -1 to 1, and values significantly below $\frac{-1}{N-1}$ indicate negative spatial autocorrelation and values significantly above $\frac{-1}{N-1}$ indicate positive spatial autocorrelation [13].

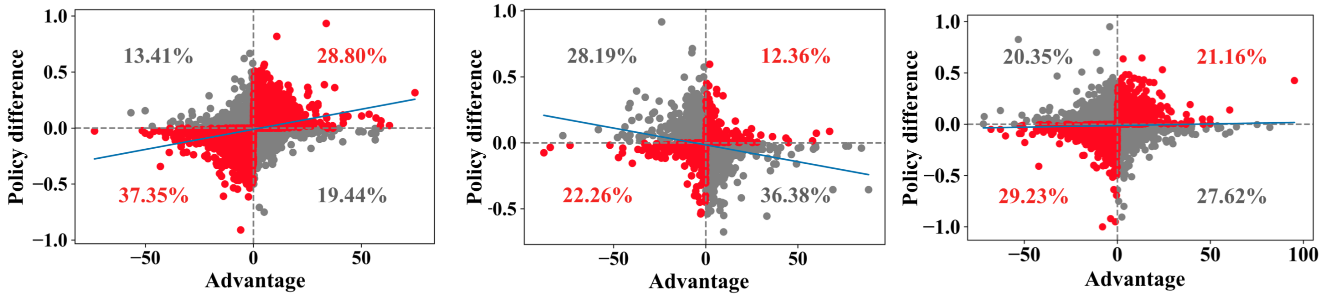
To verify if there is a significant spatial auto-correlation in the data, a hypothesis test is conducted, the null hypothesis H_0 is the values are spatially independent and assigned at random among the regions, while the alternative hypothesis H_1 is the values are spatially correlated. The null hypothesis is rejected if the statistical significance (p-value) of a Moran's I score is below a given threshold. It can be calculated through estimating the distribution of z-score of I.

The average I score of $V(s)$ and $D(s)$ among the drivers is 0.5241 and 0.5551. Given the confidence level 0.95 ($p < 0.05$), for $V(s)$, the values from 99.25% drivers reject the null hypothesis, which means $V(s)$ has spatial correlation; and for $D(s)$, the values from 99.07% drivers reject the null hypothesis, which indicates $D(s)$ also has strong spatial correlation.

(2) Integrating spatial auto-correlation in advantage correlation analysis. From the results above, it is safe to conclude that both $D(s)$ and $V(s)$ exhibit spatial auto-correlations under the weight matrix designed. Thus we should take the spatial auto-correlation into account to reduce the bias. The spatial normalized value $SN(x)$ can be calculated by Eq. 17

$$SN(x_i) = \alpha x_i + (1-\alpha)\sum_{j \neq i} w_{ij} x_j, \tag{17}$$

where x is either $D(s)$ or $V(s)$. $SN(x_i)$ is a convex combination of x_i and weighted sum of that from its neighboring cells, with combination parameter $\alpha \in [0,1]$. In this study, we employ $\alpha = 0.5$.



(a) Trending-up drivers

(b) Trending-down drivers

(c) Stabilized drivers

Figure 10: Policy difference VS. Advantage

6 EVALUATION

In this section, we apply the proposed analysis on the aforementioned real world taxi trajectory data from Shenzhen, China, to validate the established hypotheses of this paper. We quantitatively evaluate the correlations between the advantage and the policy difference among the different groups of drivers and present a case study to show that how typical drivers learn experiences in a paradigm which is similar to reinforcement learning (RL). We have released the code and data for reproducibility [2].

6.1 Experiment Settings

Following the steps discussed in Section 2 and extracting trips of taxi drivers, we use 6 months trajectory data in 2016, i.e., 07/2016-12/2016, with an average of around 600k trips per day. We conduct the experiments in two different time intervals respectively: 6am-4pm (day-time driver working hours) and 4pm-12am (night-time driver working hours). After eliminating those taxis whose records are not complete during these 6 months, there are 2,760 valid taxis found in 6am-4pm time interval, while 2,403 found in 4pm-12am time interval.

We apply Pearson's correlation coefficient and Spearman's rank correlation coefficient for the correlation analysis between policy difference and the advantage, and evaluate the statistical significance of the correlations to test our hypotheses.

Table 2: Results of correlation analysis

	Trending-up drivers	Trending-down drivers	Stabilized Drivers
Pearson's Corr	0.26	-0.21	0.023
Pearson's p-value	$6.59e^{-}12$	$2.30e^{-}25$	0.11
Spearman's Rank Corr	**0.39**	**-0.32**	0.029
Spearman's p-value	$1.17e^{-26}$	$6.85e^{-65}$	0.05

6.2 Correlation analysis

In this section, we present the correlation results between the policy difference and the advantage for each of the three groups of taxi drivers. To reduce bias in the analysis, we only consider grids (i.e., states) with sufficient visits in the data. Here we set 20 as the minimum visit count threshold, and exclude grids with fewer visits. As discussed in Section 5 for each driver we calculate the advantage over each state-action pair in a time slot T_0 and the policy difference of the same state-action pair in the next time slot T_1 over T_0 to understand how they adjust their strategies based on historical experiences. Here we use 3 weeks as the length of each time slot since it may take certain time for the adjustments to be observed.

Analysis Results. Fig. 10 shows the results of the three groups drivers, respectively. Each point in the plot represents the policy difference and advantage of a state-action pair of one driver. The x-axis is the advantage in the first time span T_0, and the y-axis is the preference difference between T_0 and T_1. The blue line is the linear regression line of the points.

Fig. 10a shows the results for the **trending-up drivers**. There is a positive correlation between the policy difference and the advantage, which imply the state-action pairs with larger advantages tend to have larger policy difference. In other words, the drivers are leaning towards increasing the relative visitation frequency of an state-action pair if she found that the advantage of the state-action pair was large in the previous time slot, and vise versa.

Fig. 10b shows the results of the **trending-down** drivers. The linear regression line of the points (blue) has a negative slope. It shows that the trending-down drivers have a negative correlation between the policy difference and the advantage, which states these drivers increase the relative visitation frequency of those state-action with smaller advantages, and vise versa, which is a counter-act with the learning process of policy-gradient RL.

Fig. 10c shows the results of the **stabilized drivers**. The slope of the linear regression line is close to 0. The stabilized drivers reflect little correlation between the policy difference and the advantage. We consider that these drivers have finished the learning process and reached a stable status.

Table 2 provides the quantitative results of three groups of taxi drivers. For **trending-up drivers**, the Spearman's rank correlation coefficient is 0.39 with a p-value of $1.17e^{-26}$, which means that the correlation between the policy difference and the advantage is significantly positive. A similar conclusion is drawn based on the result of the Pearson's correlation coefficient. Although the Pearson's correlation coefficient is smaller, it still suggests a significant positive correlation. For the **trending-down** drivers, the Spearman's rank correlation coefficient is -0.32 with a p-value of $6.85e^{-65}$. It implies that the correlation between the policy difference and the advantage is significantly negative. The Pearson's correlation analysis results suggest the same conclusion. **Stabilized drivers** have little correlation between the policy difference and the advantage with the Spearman's rank correlation coefficient of 0.029 and a p-value of 0.05.

Correlation Analysis Summary: The trending-up drivers who improved earning efficiencies over time show a similar learning process as that of the agent in an policy gradient RL algorithm, while the trending-down drivers who worsened earning efficiencies show an opposite learning process to the learning process of the agent in a policy gradient RL algorithm. This in turn proves that (1) the trending-up taxi drivers are following the paradigm of RL effectively when learning strategies, and (2) drivers tend to be more successful in terms of their increasing earning efficiency if they better follow the learning process of RL.

The result of stabilized drivers implies that these taxi drivers may have found strategies that they believe to be "optimal". They are loyal to the strategies and not temporally affected by the advantages. They are similar as agents in RL that have already reached the optimal status.

Further Investigations. Results in Fig. 10a suggest that even trending-up drivers may not follow the learning process of RL in all the scenarios. Red points represent state-action pairs on which trending-up drivers are likely to following RL, while grey points represent the scenarios that RL is not followed. We further analyze how drivers choose the scenarios (i.e., state-action pairs) to do RL. To do so, we calculate the $Q(s, a)$ of each data point in Fig. 10a, and draw histograms for the red and grey points, respectively. Fig. 11 shows the count of state-action pairs with respect to each range of $Q(s, a)$ for the trending-up drivers. Results show that trending-up drivers tend to do more RL on state-action pairs with low-to-median $Q(s, a)$ since the count of red points is significantly higher than that of the grey points for low and medium $Q(s, a)$ bins. They do not appear to do RL on state-action pairs with very high $Q(s, a)$. An explanation could be that state-action pairs with very high expected rewards are limited and well-known. The drivers' strategies on these scenarios are already pretty much optimal and can't be improved. Therefore drivers focus on improving their strategies over low or medium $Q(s, a)$ scenarios to explore new routes and patterns.

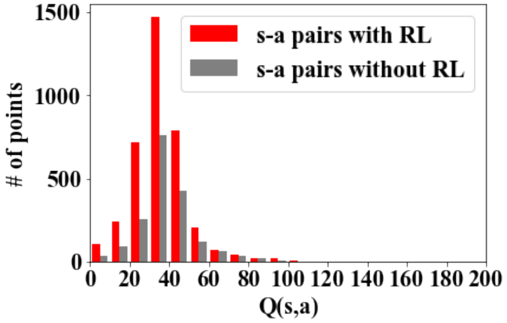
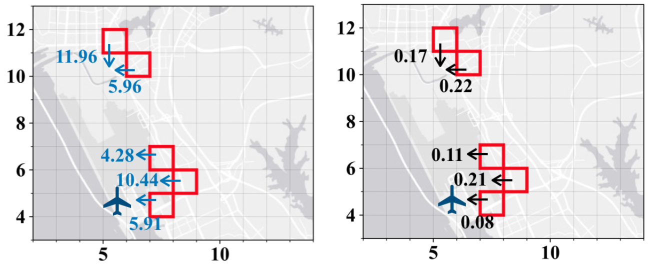


Figure 11: Distributions of s-a pairs w.r.t. $Q(s, a)$

6.3 Case Study

In this section, we provide two concrete examples from two real drivers to help illustrate our findings in details.

(1) *A trending-up driver.* We select a driver, Mike, from the group of trending-up. Mike's earning efficiency shows a monotonic increasing trend from the first week in 07/16 to the last week in 12/16. We extracted the top 5 grids with the highest visitation frequency of Mike during July and August, as shown in Fig. 12a. We can see that Mike likes working near the Airport. We calculated the advantage of the state-action pairs of these 5 grids. The state-action pair with the highest advantage value among the state-action pairs of each state is marked with a blue arrow in Fig. 12a. These blue arrows show that the driver tends to get closer to the airport to get better earnings. Then, we extract the policy difference of these state-action pairs from July to August, and the state-action pair with the largest policy difference among the state-action pairs in each state are marked with black arrows in Fig. 12b. Comparing Fig. 12b with Fig. 12a, we can find that from July to August Mike increased the probability of taking those exact actions which he learned to have the highest advantage based on experiences from July. Mike maintained a similar strategy as the agent in RL, which helped him improve his earning efficiency from July to August.



(a) State-action pairs with greatest advantages in July

(b) Greatest policy differences between July and August

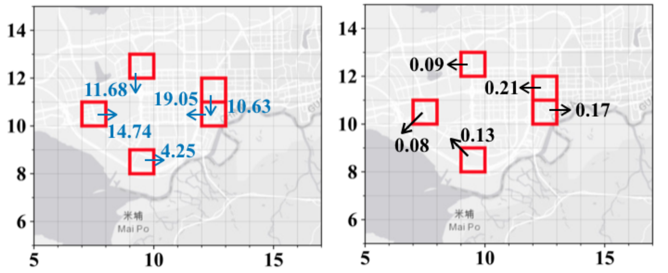
Figure 12: The learning process of Mike

(2)*A trending-down driver.* We select another driver from the group of trending-down, namely, "Jacob". Jacob's earning efficiency shows a monotonic decreasing trend from the first week in 07/16 to the last week in 12/16. We extracted the top 5 grids with the highest visitation frequency of Jacob during July and August, as shown in Fig. 13a. Jacob likes working near the downtown area. Similarly, we calculate the advantages and the policy difference for each state-action pair in these grids. The results are marked in Fig. 13a & 13b. Comparing the results in these two figures, we can find that from July to August Jacob increased the probability of taking those actions that didn't give him high advantages in July. It is the opposite to what an agent in RL would do. Thus, the earning efficiency of Jacob was lowered from July to August.

6.4 Takeaways and Discussions

Based on our study upon a large real-world taxi trajectory dataset, we acquired promising findings about whether a taxi driver follows the learning process of RL and why different groups of taxi drivers have different earning efficiency trends over time. The takeaways are summarized:

(1) Taxi drivers, especially the ones with improving earning efficiencies, indeed follow the learning process of RL. Drivers with



(a) State-action pairs with greatest advantages in July

(b) Greatest policy differences between July and August

Figure 13: The learning process of Jacob

the different trends of earning efficiency result from the different extents to follow the paradigm of RL.

(2) Even the best drivers cannot completely follow the RL paradigm in all the scenarios. The possible reasons are that human drivers have limited memories and they do not precisely calculate the advantage over all the state-action pairs. Trending-up drivers tend to better follow the RL paradigm for those state-action pairs with low-to-medium expected rewards. The reason could be that their strategies are already (near) optimal for those high-reward state-action pairs. The improvement primarily comes from the low-to-medium reward scenarios.

Our findings establish the foundation for future research related to behavior analysis of taxi drivers. It can be used for strategy recommendations. For example, for slow-growing drivers, one can focus on helping them keep better track of their advantages so that they better follow RL and their earning efficiencies grow faster. Also, one can expect drivers to learn the best strategies in the most profitable areas quickly. Learning is efficient if drivers focus more on improving their decisions in low-to-medium reward areas.

7 RELATED WORK

Taxi operating strategies (e.g., dispatching, passenger seeking), and driver behavior analysis have been extensively studied in recent years due to the emergence of the ride-sharing business model and urban intelligence. The related works are summarized below.

Urban Computing integrates urban sensing, data management, and data analytic as a unified process to explore, analyze, and solve problems related to people's everyday life [7, 10–12, 28, 30, 33, 34, 38, 53, 56, 58, 60]. In particular, a group of works have studied the topic of taxi operation [8, 9, 21, 32, 35, 42, 46, 51], such as vehicle dispatching with reinforcement learning [17, 18, 23, 24, 27, 39, 43, 46, 49, 61], and passenger-seeking strategies [14, 19, 36, 54, 55, 57]. They aim to find optimal solutions to improve the revenue of individual taxi drivers as well as the entire fleet. For instance, [41] solved the passenger-seeking problem by giving direction recommendations to drivers. However, few studies investigate the relation between the machine learned strategies and human drivers' strategies. Some studies directly assume that human drivers follow reinforcement learning [37, 52, 62] without validation through real cases. To the best of our knowledge, *our study makes the first attempt to validate if taxi drivers follow the paradigm of reinforcement learning when earning their driving experiences.*

Human Learning is a process of interacting between a person and the external environment, which leads human to change capacity permanently not due to biological maturation [20]. To characterize how the process works, research in Cognitive Neuroscience, Psychological Sciences, and Behavioural Sciences has studied over five decades [26]. [3] investigated the role of brain's modular structures and found that flexibility measured by the allegiance of nodes to modules in a past session could predict the relative amount of learning in a future session. [45] contended the essential factors that can lead to progress in learning mathematics from the perspective of psychology. [40] introduced a structured learning tool and teaching process to translate the learning principles into practice for learning clinical skills regarding behavioral sciences. Compared with previous works, *we deliver an innovative insight of leveraging the understanding of human learning to engineer the learning process through machine learning.*

8 CONCLUSION

Previous works make an assumption that human learners follow the paradigm of reinforcement learning (RL) to change their strategies. We propose a novel framework, including trending analysis, learning modeling, and strategy validation, to validate this assumption. Our experiments on a large-scale real-world taxi trajectory data prove that the taxi drivers' strategy change follows the learning process of RL and the drivers with different trends of earning efficiency have the different extents to follow RL. Our framework and findings provide an important sight in the fields of human behavior learning and taxi operation management.

ACKNOWLEDGMENTS

Menghai Pan, Weixiao Huang, and Yanhua Li were partly supported by NSF grants IIS-1942680 (CAREER), CNS-1657350 (CRII), and CMMI-1831140. Xun Zhou was partially supported by a grant from the SAFER-SIM University Transportation Center. Zhenming Liu was supported by NSF grants NSCI-1835821 and IIS-1755769 (CRII).

REFERENCES

[1] OpenStreetMap. http://www.openstreetmap.org/.
[2] Project code and dataset. https://github.com/paperpublicsource/learning_strategy.
[3] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 2011.
[4] R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.
[5] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.
[6] G. W. Corder and D. I. Foreman. *Nonparametric statistics: A step-by-step approach.* John Wiley & Sons, 2014.
[7] Y. Ding, Y. Li, K. Deng, H. Tan, M. Yuan, and L. M. Ni. Detecting and analyzing urban regions with high impact of weather change on transport. *IEEE Transactions on Big Data*, 2016.
[8] X. Dong, M. Zhang, S. Zhang, X. Shen, and B. Hu. The analysis of urban taxi operation efficiency based on gps trajectory big data. *Physica A: Statistical Mechanics and its Applications*, 528:121456, 2019.
[9] Y. Duan, N. Wang, and J. Wu. Optimizing order dispatch for ride-sharing systems. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2019.
[10] J. Fan, Y. Li, Y. Liu, Y. Zhang, and C. Ma. Analysis of taxi driving behavior and driving risk based on trajectory data. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 220–225. IEEE, 2019.
[11] Z. Fan, Q. Chen, R. Jiang, R. Shibasaki, X. Song, and K. Tsubouchi. Deep multiple instance learning for human trajectory identification. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 512–515, 2019.

[12] J. C. Gareau, É. Beaudry, and V. Makarenkov. An efficient electric vehicle path-planner that considers the waiting time. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 389–397, 2019.

[13] A. Getis and J. K. Ord. The analysis of spatial association by use of distance statistics. *Geographical analysis*, 24(3):189–206, 1992.

[14] S. Guo, C. Chen, J. Wang, Y. Liu, X. Ke, Z. Yu, D. Zhang, and D.-M. Chiu. Rod-revenue: Seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data. *IEEE Transactions on Mobile Computing*, 2019.

[15] K. H. Hamed and A. R. Rao. A modified mann-kendall trend test for autocorrelated data. *Journal of hydrology*, 204(1-4):182–196, 1998.

[16] J. D. Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, NJ, 1994.

[17] S. He and K. G. Shin. Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination. In *The World Wide Web Conference*, pages 2806–2813, 2019.

[18] J. Holler, R. Vuorio, Z. Qin, X. Tang, Y. Jiao, T. Jin, S. Singh, C. Wang, and J. Ye. Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1090–1095. IEEE, 2019.

[19] Z. Huang, J. Tang, G. Shan, J. Ni, Y. Chen, and C. Wang. An efficient passenger-hunting recommendation framework with multitask deep learning. *IEEE Internet of Things Journal*, 6(5):7713–7721, 2019.

[20] K. Illeris. *How we learn: Learning and non-learning in school and beyond*. London/New York: Routledge, 2007.

[21] J. Yuan, Y. Zheng, L. Zhang, X. Xie. T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2390–2403, 2013.

[22] B. James, K. L. James, and D. Siegmund. Tests for a change-point. *Biometrika*, 74(1):71–83, 1987.

[23] J. Jin, M. Zhou, W. Zhang, M. Li, Z. Guo, Z. Qin, Y. Jiao, X. Tang, C. Wang, J. Wang, et al. Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1983–1992, 2019.

[24] J. Ke, F. Xiao, H. Yang, and J. Ye. Optimizing online matching for ride-sourcing services with multi-agent deep reinforcement learning. *arXiv preprint arXiv:1902.06228*, 2019.

[25] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[26] G. R. Lefrancois. *Theories of human learning*. Cambridge University Press, 2019.

[27] M. Li, Z. Qin, Y. Jiao, Y. Yang, J. Wang, C. Wang, G. Wu, and J. Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*, pages 983–994, 2019.

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

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

[30] C. Liu, K. Deng, C. Li, J. Li, Y. Li, and J. Luo. The optimal distribution of electric-vehicle chargers across a city. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 261–270. IEEE, 2016.

[31] L. Liu, C. Andris, A. Biderman, and C. Ratti. Revealing taxi driver's mobility intelligence through his trace. In *Movement-Aware Applications for Sustainable Mobility: Technologies and Approaches*, pages 105–120. IGI Global, 2010.

[32] Z. Liu, Z. Gong, J. Li, and K. Wu. Mobility-aware dynamic taxi ridesharing. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 961–972. IEEE, 2020.

[33] B. Lyu, S. Li, Y. Li, J. Fu, A. C. Trapp, H. Xie, and Y. Liao. Scalable user assignment in power grids: a data driven approach. In *SIGSPATIAL GIS*. ACM, 2016.

[34] M. Qu, H. Zhu, J. Liu, G. Liu, H. Xiong. A Cost-Effective Recommender System for Taxi Drivers. In *The 20th International Conference on Knowledge Discovery and Data Mining (SIGKDD'14)*, pages 45–54, New York, NY, 2014. ACM.

[35] Q. Ma, Z. Cao, K. Liu, and X. Miao. Qa-share: Toward an efficient qos-aware dispatching approach for urban taxi-sharing. *ACM Transactions on Sensor Networks (TOSN)*, 16(2):1–21, 2020.

[36] M. Pan, W. Huang, Y. Li, X. Zhou, Z. Liu, R. Song, H. Lu, Z. Tian, and J. Luo. Dhpa: Dynamic human preference analytics framework: A case study on taxi drivers' learning curve analysis. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–19, 2020.

[37] M. Pan, Y. Li, X. Zhou, Z. Liu, R. Song, and J. Luo. Dissecting the learning curve of taxi drivers: A data-driven approach. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019.

[38] Y. Pang, K. Tsubouchi, T. Yabe, and Y. Sekimoto. Replicating urban dynamics by generating human-like agents from smartphone gps data. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 440–443, 2018.

[39] Z. Qin, J. Tang, and J. Ye. Deep reinforcement learning with applications in transportation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3201–3202, 2019.

[40] I. Rolfe and R. Sanson-Fisher. Translating learning principles into practice: a new strategy for learning clinical skills. *Medical education*, 36(4):345–352, 2002.

[41] H. Rong, X. Zhou, C. Yang, Z. Shafiq, and A. Liu. The rich and the poor: A markov decision process approach to optimizing taxi driver revenue efficiency. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2329–2334. ACM, 2016.

[42] W. S. Ma, Y. Zheng. A large-scale dynamic taxi ridesharing service. In *The 29th International Conference on Data Engineering (ICDE'13)*, pages 410–421, New York, NY, 2013. IEEE.

[43] J. Shi, Y. Gao, W. Wang, N. Yu, and P. A. Ioannou. Operating electric vehicle fleet for ride-hailing services with reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[44] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[45] R. Skemp. *The psychology of mathematics learning: Expanded American edition*. Hillsdale, New Jersey: Erlbaum, 1987.

[46] Y. Song, N. Sun, and H. Chen. Demand adaptive multi-objective electric taxi fleet dispatching with carbon emission analysis. In *2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 1–5. IEEE, 2019.

[47] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[48] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[49] X. Tang, Z. Qin, F. Zhang, Z. Wang, Z. Xu, Y. Ma, H. Zhu, and J. Ye. A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1780–1790, 2019.

[50] L. Wang, W. Zhang, X. He, and H. Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2447–2456, 2018.

[51] Z. Wang. Taxi scheduling optimization with incomplete information. In *2019 International Conference on Information Technology and Computer Application (ITCA)*, pages 266–270. IEEE, 2019.

[52] G. Wu, Y. Li, J. Bao, Y. Zheng, J. Ye, and J. Luo. Human-centric urban transit evaluation and planning. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 547–556. IEEE, 2018.

[53] T. Xu, H. Zhu, X. Zhao, Q. Liu, H. Zhong, E. Chen, and H. Xiong. Taxi driving behavior analysis in latent vehicle-to-vehicle networks: A social influence perspective. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294. ACM, 2016.

[54] Y. Ge and H. Xiong and A. Tuzhilin and K. Xiao and M. Gruteser. An energy-efficient mobile recommender system. In *The the 16th International Conference on Knowledge Discovery and Data Mining*, pages 899–908, New York, NY, 2010. ACM.

[55] Y. Ge, C. Liu, H. Xiong, J. Chen. A Taxi Business Intelligence System. In *The 17th International Conference on Knowledge Discovery and Data Mining*, pages 735–738, New York, NY, 2011. ACM.

[56] G. P. Yatnalka and H. S. Narman. A matching model for vehicle sharing based on user characteristics and tolerated-time. In *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT)*, pages 143–147. IEEE, 2019.

[57] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun. Where to find my next passenger. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 109–118, New York, NY, 2011. ACM.

[58] Z. Yuan, X. Zhou, and T. Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 984–992. ACM, 2018.

[59] C. Zeng and N. Oren. Dynamic taxi pricing. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014), Prague, Czech Republic*, pages 1135–1136. ECAI 2014.

[60] C. Zhang, Y. Li, J. Bao, S. Ruan, T. He, H. Lu, Z. Tian, C. Liu, C. Tian, J. Lin, et al. Effective recycling planning for dockless sharing bikes. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 62–70, 2019.

[61] M. Zhou, J. Jin, W. Zhang, Z. Qin, Y. Jiao, C. Wang, G. Wu, Y. Yu, and J. Ye. Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2645–2653, 2019.

[62] X. Zhou, H. Rong, C. Yang, Q. Zhang, A. V. Khezerlou, H. Zheng, M. Z. Shafiq, and A. X. Liu. Optimizing taxi driver profit efficiency: A spatial network-based markov decision process approach. *IEEE Transactions on Big Data*, 2018.