2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Transforming Policy via Reward Advancement

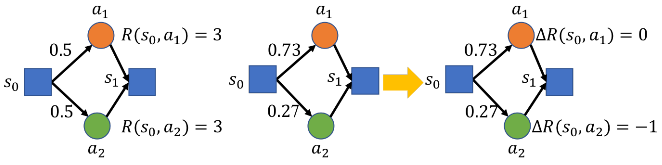
Guojun Wu[1], Yanhua Li[1] and Jun Luo[2]

***Abstract*— Many real world human behaviors can be characterized as sequential decision making processes, such as urban travelers' choices of transport modes and routes [1]. Differing from choices controlled by machines, which in general follows *perfect rationality* to adopt the policy with highest reward, studies have revealed that human agents make sub-optimal decisions under *bounded rationality* [2]. Such behaviors can be modeled using maximum causal entropy (MCE) principle [3]. In this paper, we define and investigate a novel reward transformation problem (namely, *reward advancement*): Recovering the range of additional reward functions that transform the agent's policy from π_o to a predefined target policy π_t under MCE principle. We show that given an MDP and a target policy π_t, there are infinite many additional reward functions that can achieve the desired policy transformation. Moreover, we propose an algorithm to further extract the additional rewards with minimum "cost" to implement the policy transformation. We demonstrated the correctness and accuracy of our reward advancement solution using both synthetic data and a large-scale (6 months) passenger-level public transit data from Shenzhen, China.**

I. Introduction

In sequential decision making problems [3], human agents complete tasks by evaluating the rewards received over states traversed and actions employed. Each human agent may have her own unique reward function, which governs how much reward she may receive over states and actions [4], [5]. For example, urban travelers may evaluate the travel cost vs travel time with different weights, when deciding which transport mode, route, and transfer stations to take [1]. Uber drivers may prefer different urban regions to look for passengers, depending on their familiarity to the regions, and distance to their home locations, etc [6]. To quantify and measure the unique reward function each human agent possesses, maximum causal entropy inverse reinforcement learning (IRL) [7] has been proposed to find the reward function and the corresponding policy, that best represents demonstrated behaviors from the human agent with the highest causal entropy, subject to the constraint of matching feature expectations to the distribution of demonstrated behaviors.

Going beyond the human agent reward learning problem, in this paper, we move one step further to investigate how we can influence and change agent's policy (i.e., decisions) to a target policy π_t from the original policy π_o observed from the agent's trajectories, by purposely updating and advancing the rewards received by the human agent. Figure 1 illustrates this problem with a concrete example. A small scale Markov Decision Process (MDP) has two states $\{s_0, s_1\}$, and two actions $\{a_1, a_2\}$. Starting from s_0, an agent can reach s_1 by either taking action a_1 or a_2. The original rewards received by the agent from taking action a_1 are a_2 are equal, i.e., $R(s_0, a_1) = R(s_0, a_2) = 3$ (Figure 1(a)). As a result, the policy of choosing a_1 vs a_2 are both 50% by the maximum entropy principle [7]. If we want the human agent to switch to a new policy (see Figure 1(b)) with $\pi_t(a_1|s_0) = 0.73$ and $\pi_t(a_2|s_0) = 0.27$, respectively, we can introduce additional reward $\Delta R(s_0, a_2) = -1$ to state-action (s_0, a_2), and keep $R(s_0, a_1)$ invariant (Figure 1(c)).



(a). Original Reward and Policy (b). Target Policy (c). Additional Reward

Fig. 1. An Example of Reward Advancement

This problem of finding additional reward to transform human agent's policy with minimum cost is of great practical importance. For example, urban passengers employ their own unique policies to choose transit modes and transfer stations, which may collectively lead to unbalanced crowd flows, i.e., under- and over-supplied traffic over stations and routes. One way to mitigate such a problem is to motivate or incentivize passengers to autonomously change and transform their policies, e.g., by providing additional reward to the those passenger agents, in forms of coupons, discounted price, etc [8], [9]. Moreover, it is crucial how to achieve this goal with minimum overall cost. In the literature, reward transformations [10], [11] have been studied extensively, primarily focusing on transforming the reward, with the goal of preserving the same policy (which is formally termed as "reward shaping"). Differing from reward shaping, our design goal is more general, namely, transforming rewards, so the agent behaves as a target policy π_t, which may or may not be the same as the agent's original policy π_o. We refer this problem as a "reward advancement" problem.

In this paper, we make the first attempt to tackle the reward advancement problem. Given a Markov Decision Process and a target policy π_t, we investigate the range of additional rewards that can transform the agent's policy to the predefined target policy π_t under MCE principle. Our main contributions are summarized as follows.

- We are the first to define and study the reward advancement problem, namely, finding the updating rewards to transform human agent's behaving policy to a prede-

[1] Guojun Wu and Yanhua Li are with Worcester Polytechnic Institute (WPI), USA, {gwu, yli15}@wpi.edu.

[2] Jun Luo is with Machine Intelligence Center (Hong Kong), Lenovo Group Limited, jluo1@lenovo.com.

978-1-7281-1398-2/19/$31.00 ©2019 IEEE

fined target policy. We provide a close-form solution to this problem. The solution indicates that there exist infinite many such additional rewards, that can achieve the desired policy transformation.
- Moreover, we define and investigate min-cost reward advancement problem, which aims to find the additional rewards that can transform the agent's policy to π_t, while minimizing the cost of the policy transformation.
- We also demonstrated the correctness and accuracy of our reward advancement algorithm using both synthetic data and a large-scale (6 months) passenger-level public transit data from Shenzhen, China.

The paper is organized as follows, Section II discusses preliminaries and formally defines the reward advancement problem. Section III introduces our maximum entropy reward advancement algorithm. Section V presents evaluation results using both grid world scenario and real world urban passenger data. Section VI concludes the paper.

II. Preliminaries and Problem Definition

In this section, we review the basics of finite Markov Decision Process and Maximum Causal Entropy (MCE) policy.

A. Markov Decision Process

An MDP is represented as a tuple $\langle S, A, T, \gamma, \mu_0, R\rangle$, where S is a finite set of states and 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], $\mu_0 : S \to [0,1]$ is the initial distribution, and $R : S \times A \to \mathbf{R}$ 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 $\zeta = [s_0, a_0, s_1, a_1, \ldots, s_L, a_L]$ to denote a trajectory generated by the Markov Decision Process (MDP), which is a sequence of state-action pair. L is the length of trajectory. The planning problem in an MDP aims to find a policy π, such that the expected total reward is maximized.

B. Policy under Maximum Causal Entropy Principle

One well-known solution to the inverse reinforcement learning problem is Maximum Causal Entropy Inverse Reinforcement Learning [3]. It proposes to find the policy that best represents demonstrated behaviors with highest causal entropy $H(A||S)$, which is calculated by $H(A||S) = -\sum_{s\in S}\sum_{a\in A} D(s,a)\ln \pi(a|s)$, where $D(s,a)$ represents the expected visitation frequency of the state-action pair (s,a), when one trajectory is generated under policy $\pi(a|s)$.

The policy under maximum causal entropy principle (i.e., *MCE policy*) best represents the demonstrated behaviors with the highest causal entropy, and is subject to matching the reward expectations of demonstrated behaviors. Denote $Q(s,a) = R(s,a) + \sum_{s'\in S} T(s'|s,a)\sum_{a'\in A}\pi(a'|s')Q(s',a')$ as Q-function on state-action pair (s,a), indicating the expected rewards to be received starting from (s,a), MCE policy is

$$\pi(a|s) = \frac{e^{Q(s,a)}}{\sum_{a'\in A} e^{Q(s,a')}}. \quad (1)$$

Eq.(1) is the policy conducted by the human agent, that best matches her generated trajectory data. Usually, the $Q(s,a)$ can be represented using a parameterized function $Q(s,a|\theta)$, for instance, a neural network model. If we use $\tilde{TR}$ to represent expert trajectories we collected, the Q-function $Q(s,a|\theta)$'s then can be estimated by solving a maximum likelihood estimation problem,

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, L(\theta) = \operatorname{argmax}\theta \sum_{\zeta\in \tilde{TR}} \ln P(\zeta|\theta). \quad (2)$$

III. Reward Advancement

Inverse reinforcement learning problem [10], [3], [7], [12], [13], [14], [15] aims to inversely learn agent's reward (or preference) function from their demonstrated trajectories, namely, inferring how agent makes decisions. In this work, we move one step further to investigate how we can influence and transform agent's decision-making policy to a target policy π_t from the original policy π_o observed from the demonstrated trajectories, by purposely updating and advancing the reward functions $R(s,a)$ in the MDP. Reward transformations [10], [16] have been studied in the literature, primarily focusing on transforming the rewards, with the goal of preserving the same policy (which is formally termed as "reward shaping"). Differing from reward shaping, our design goal is more general, say, transforming rewards, so the agent behaves as a predefined target policy π_t, which may or may not be the agent's current policy π_o. This problem is referred to as a "reward advancement" problem, and we formally define it as follows.

Reward Advancement Problem. Given an MDP $\langle S, A, T, \mu_0, R_o\rangle$, the agent's MCE policy is π_o. we aim to find additional rewards ΔR to be added to the original reward R_o, such that the agent's MCE policy under the updated MDP $\langle S, A, T, \mu_0, R_o + \Delta R\rangle$ follows a predefined target policy π_t. Without loss of generality, we use $\gamma = 1$ as discount factor for simplicity.

For MDP $\langle S, A, T, \mu_0, R_o\rangle$, each policy π running on it leads to a unique Q-function:

$$Q^\pi_o(s,a) = R_o(s,a) + \sum_{s'\in S} T(s'|s,a)\sum_{a'\in A}\pi(s',a')Q_o(s',a').$$

From Maximum Causal Entropy Inverse Reinforcement Learning, there exists a unique MCE policy π_o in form of eq.(1) that maximizes the likelihood of observing the given demonstration data. However, when appropriate additional rewards $\Delta R(s,a)$ are provided, MDP becomes $\langle S, A, T, \mu_0, R_o + \Delta R\rangle$, and the underlying MCE policy may change to π. This occurs because the additional rewards ΔR transforms and advances the MCE policy from π_o to π. In this case, the Q-function with MCE policy π is characterized as $Q^\pi(s,a) = R(s,a) + \sum_{s'\in S} T(s'|s,a)\sum_{a'\in A}\pi(a'|s')Q^\pi(s',a')$,

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

where $R(s,a) = R_o(s,a) + \Delta R(s,a)$, $Q^{\pi}(s,a) = Q_o^{\pi}(s,a) + \Delta Q(s,a)$, and $\Delta Q(s,a) = \Delta R(s,a) + \sum_{s' \in S} T(s'|s,a) \sum_{a' \in A} \pi(a'|s') \Delta Q(s',a')$.

As a result, transforming from the original MCE policy π_o, the new MCE policy π is a function of addition reward ΔR, or equivalently ΔQ, i.e., $\pi(a|s;\Delta Q)$. Given a predefined π_t, finding the right ΔQ, such that $\pi(a|s;\Delta Q) = \pi_t(a|s)$ for any $s \in S$ and $a \in A$, solves the reward advancement problem. The following Theorem 1 introduces the complete solution set to this problem.

Theorem 1. *Given an MDP $\langle S, A, T, \mu_0, R_o \rangle$, the sufficient and necessary condition to transform its MCE policy to a predefined policy π_t is to provide additional Q-function ΔQ, such that*

$$\Delta Q(s,a) = \ln \frac{\pi_t(a|s)}{e^{Q_o^{\pi_t}(s,a)}} + \beta(s), \tag{3}$$

where $\beta : S \to \mathbb{R}$ is any real number function defined on states. Such additional Q-function is called "advancement function".

Proof. (Sufficiency.) If $\Delta Q(s,a)$ follows eq.(3), the new Q-function over π_t becomes:

$$Q^{\pi_t}(s,a) = Q_o^{\pi_t}(s,a) + \Delta Q(s,a) = \ln \pi_t(a|s) + \beta(s).$$

As a result, for any (s,a), the ratio between $e^{Q^{\pi_t}(s,a)}$ and $\sum_{a'} e^{Q^{\pi_t}(s,a')}$ is exactly $\pi_t(a|s)$, thus π_t is the MCE policy of the new MDP.

(Necessity.) Given a certain additional Q-function ΔQ, if it transforms the MCE policy to π_t, that infers

$$\pi_t(a|s) = \frac{e^{Q_o^{\pi_t}(s,a) + \Delta Q(s,a)}}{\sum_{a' \in A} e^{Q_o^{\pi_t}(s,a') + \Delta Q(s,a')}}, \tag{4}$$

$$e^{\Delta Q(s,a)} = \frac{\pi_t(a|s)}{e^{Q_o^{\pi_t}(s,a)}} \sum_{a' \in A} e^{Q_o^{\pi_t}(s,a') + \Delta Q(s,a')}. \tag{5}$$

Define $\beta(s) = \ln \sum_{a' \in A} e^{Q_o^{\pi_t}(s,a') + \Delta Q(s,a')}$, we have $\Delta Q(s,a) = \ln \frac{\pi_t(a|s)}{e^{Q(s,a)}} + \beta(s)$. It completes the proof. □

The advancement function introduced in Theorem 1 is defined on additional Q-function, which can be easily "translated" into additional reward function ΔR by the following mapping function.

$$\Delta R(s,a) = \Delta Q(s,a) - \sum_{s' \in S} T(s'|s,a) \sum_{a' \in A} \pi_t(s',a') \Delta Q(s',a'). \tag{6}$$

Theorem 1 indicates that there are infinite many advancement functions that can transform an original MCE policy π_o to a given π_t. However, different advancement functions may lead to different costs in reality to apply the additional rewards. For example, in ride-hailing service, additional rewards provided to Uber drivers could be in the form of monetary values; in urban public transportation systems, the additional rewards to passengers could be in forms of ride discount. More additional rewards applied lead to more cost to the system. Without lower bound on $\beta(s)$, the advancement function ΔQ can be as low as $-\infty$. In turn, the addition rewards ΔR inferred from eq.(6) can be arbitrarily small as well. It is equivalent to increase the ride rate to be extremely large for public transits, which is not feasible in real world scenario. Next, we will introduce and provide solution to the reward advancement problem with minimum cost as the objective.

IV. Min-Cost Reward Advancement

Now, we investigate how to identify additional rewards that transform the agent to an MCE policy π_t, while guaranteeing minimum "implementation cost", namely, a *min-cost reward advancement problem*. Without loss of generality, we consider that the total cost of transforming the agent's policy is equal to the expected additional rewards offered to the agent, i.e.,

$$\begin{aligned} C(\Delta R) &= \sum_{s \in S} \sum_{a \in A} D_t(s,a) \Delta R(s,a) \\ = C(\Delta Q) &= \sum_{s \in S} \sum_{a \in A} \mu_0(s) \pi_t(a|s) \Delta Q(s,a), \end{aligned}$$

where $D_t(s,a)$ is the state-action pair visitation frequency under target policy π_t, and $\mu_0(s)$ is the initial state distribution. As a result, the general form of min-cost reward advancement problem can be formulated as follows.

Problem 1: Min-Cost Reward Advancement:

$$\min_{\Delta Q} \quad C(\Delta Q) = \sum_{s \in S} \sum_{a \in A} \mu_0(s) \pi_t(a|s) \Delta Q(s,a), \tag{7}$$

$$s.t. \quad \pi(a|s;\Delta Q) = \pi_t(a|s), \quad \forall s \in S, a \in A, \tag{8}$$

$$\Delta Q(s,a) \geq \phi(s,a), \quad \forall s \in S, a \in A. \tag{9}$$

Constraint eq.(8) guarantees to transform the agent's MCE policy to π_t, representing the infinite many feasible solutions given in Theorem 1. Constraint eq.(9) specifies the minimum additional expected reward we can offer to the agent, namely, $\phi : S \times A \to \mathbb{R}$ are system constants. Constraint eq.(9) makes sense in reality, which infers that the expected reward received by the agent cannot be lower than a certain minimum value. without this constraint, $\Delta Q(s,a) = -\infty$ becomes a trivial solution to Problem 1.

Theorem 2. *The solution to the min-cost reward advancement problem in eq.(7)-(9) is*

$$\begin{cases} \Delta Q(s,a) = \max\limits_{a' \in A} (\ln \frac{e^{Q_o^{\pi_t}(s,a')}}{e^{Q_o^{\pi_t}(s,a)}} \frac{\pi_t(a|s)}{\pi_t(a'|s)} + \phi(s,a')), \mu_0(s) > 0, \\ \Delta Q(s,a) \geq \max\limits_{a' \in A} (\ln \frac{e^{Q_o^{\pi_t}(s,a')}}{e^{Q_o^{\pi_t}(s,a)}} \frac{\pi_t(a|s)}{\pi_t(a'|s)} + \phi(s,a')), \mu_0(s) = 0. \end{cases}$$

Proof. Theorem 1 indicates the solution set of constraint eq.(8). As a result, we can safely remove constraint eq.(8) and replace $\Delta Q(s,a)$ with eq.(3). Then, Problem 1 is transferred to the following format, with variable $\beta(s)$, instead.

Algorithm 1 Min-Cost Reward Advancement via Monte Carlo Policy Evaluation

```
1: INPUT: States $S$, Actions $A$, Original Rewards $R_o$ and Original Trajectory Set $\tilde{TR}$;
2: OUTPUT: Additional reward on each state-action pair $\Delta R(s,a)$ (One from many solutions in Theorem 2);
3: For each state-action pair $(s,a)$, calculate $Q_o^{\pi_t}(s,a) = \frac{\sum_{\zeta\in\tilde{TR}_{s,a}}\sum_{t\geq t_{(s,a)|\zeta}}\frac{\pi_t(a|s)}{\pi_o(a|s)}R_o(s_t,a_t)}{|\tilde{TR}_{s,a}|}$;
4: Calculate $\beta(s) = \max_{a\in A}(\ln\frac{e^{Q_o^{\pi_t}(s,a)}}{\pi_t(a|s)} + \phi(s,a))$ for each state $s$;
5: Calculate $\Delta Q(s,a) = \ln\frac{\pi_t(a|s)}{e^{Q_o^{\pi_t}(s,a)}} + \beta(s)$ for each stat-action pair $(s,a)$;
6: For each $(s,a)$, calculate $\Delta R(s,a) = \Delta Q(s,a) - \frac{1}{|\tilde{TR}_{s,a,s',a'}|}\sum_{\zeta\in\tilde{TR}_{s,a,s',a'}}\frac{\pi_t(a'|s')}{\pi_o(a'|s')}\Delta Q(s',a')$;
7: Return $\Delta R(s,a)$;
```

Problem 2: Min-Cost Reward Advancement in β

$$\min_{\beta} \quad \sum_{s\in S}\sum_{a\in A}\mu_0(s)\pi_t(a|s)(\beta(s)+\ln\frac{\pi_t(a|s)}{e^{Q_o^{\pi_t}(s,a)}}), \qquad (10)$$

$$s.t \quad \beta(s) \geq \max_{a\in A}(\ln\frac{e^{Q_o^{\pi_t}(s,a)}}{\pi_t(a|s)} + \phi(s,a)), \forall s\in S. \quad (11)$$

Eq.(10) is clearly a linear function of $\beta(s)$. As a result, the minimum objective function eq.(10) is achieved, when each $\beta(s)$ is minimum, that is, when the equality is attained in constraint eq.(11):

$$\beta(s) = \max_{a\in A}(\ln\frac{e^{Q_o^{\pi_t}(s,a)}}{\pi_t(a|s)} + \phi(s,a)), \quad \mu_0(s)\geq 0. \quad (12)$$

Moreover, eq.(10) indicates that the value of objective function only hinges on $\beta(s)$, with $\mu_0(s) > 0$. For other states with $\mu_0(s) = 0$, $\beta(s)$ only needs to fulfill the constraint eq.(11), and has no impact on the value of the objective function. The complete set of solutions to Problem 2 is as follows.

$$\begin{cases}\beta(s) = \max_{a\in A}(\ln\frac{e^{Q_o^{\pi_t}(s,a)}}{\pi_t(a|s)} + \phi(s,a)), & \mu_0(s) > 0,\\ \beta(s) \geq \max_{a\in A}(\ln\frac{e^{Q_o^{\pi_t}(s,a)}}{\pi_t(a|s)} + \phi(s,a)), & \mu_0(s) = 0.\end{cases}$$

Plugging the above solution set to eq.(3) yields the solution to ΔQ, and completes the proof. □

Again, the solutions to advancement function ΔQ can be mapped to additional rewards ΔR, by applying

$$\Delta R(s,a) = \Delta Q(s,a) - \sum_{s'\in S}T(s'|s,a)\sum_{a'\in A}\pi_t(a'|s')\Delta Q(s',a').$$

Moreover, Theorem 2 indicates that the optimal solutions of $\Delta Q(s,a)$ on states with $\mu_0(s) > 0$ are unique (with equalities), while solutions on states with $\mu_0(s) = 0$ are infinite many, following inequalities.

Practical challenges in algorithm design. Theorem 2 provides a nice close-form solution set to the min-cost reward advancement problem. However, it requires calculating $Q_o^{\pi_t}(s,a)$ based on original reward $R_o(s,a)$ and target policy $\pi_t(a|s)$. A natural way to calculate it is value iteration method [7], [17], which employs an iterative framework, and solves a dynamic programming sub-problem within each iteration. As a result, such approach would be time-consuming, when the state space is large. Moreover, the value iteration will not work, when the transition matrix T is unknown.

To address the problems of scalability and unknown dynamics, we propose to apply Monte Carlo policy evaluation method to estimate Q-function of original rewards under target policy $Q_o^{\pi_t}(s,a)$. Here, we adopt the first-visit Monte Carlo method [17]. From first-visit Monte Carlo method, we have $Q_o(s,a)$ as the average total reward after first visit to state-action pair (s,a) on each trajectory, denoted as

$$Q_o(s,a) = \frac{1}{|\tilde{TR}_{s,a}|}\sum_{\zeta\in\tilde{TR}_{s,a}}\sum_{t\geq t_{(s,a)|\zeta}}R_o(s_t,a_t), \qquad (13)$$

where ζ is one trajectory, (s_t,a_t) is a state-action pair on the trajectory ζ, $t_{(s,a)|\zeta}$ is the first occurrence of (s,a) in ζ and $|\tilde{TR}_{s,a}|$ is number of trajectories traversing (s,a). Since trajectory set $\tilde{TR}_{s,a}$ were collected with original policy π_o, we adopt importance sampling method to estimate Q-function of original reward under target policy $Q_o^{\pi_t}(s,a)$ by

$$Q_o^{\pi_t}(s,a) = \frac{\sum_{\zeta\in\tilde{TR}_{s,a}}\sum_{t\geq t_{(s,a)|\zeta}}\frac{\pi_t(a|s)}{\pi_o(a|s)}R_o(s_t,a_t)}{|\tilde{TR}_{s,a}|}, \quad (14)$$

where the $\frac{\pi_t(a|s)}{\pi_o(a|s)}$ is the importance ratio. Similarly, we can estimate $\Delta R(s,a)$ through importance sampling as

$$\begin{aligned}\Delta R(s,a) &= \Delta Q(s,a)\\ &- \frac{1}{|\tilde{TR}_{s,a,s',a'}|}\sum_{\zeta\in\tilde{TR}_{s,a,s',a'}}\frac{\pi_t(a'|s')}{\pi_o(a'|s')}\Delta Q(s',a'), \quad (15)\end{aligned}$$

where $\tilde{TR}_{s,a,s',a'}$ are trajectories containing (s,a,s',a') and $|\tilde{TR}_{s,a,s',a'}|$ is the number of those trajectories. Moreover, if the original policy π_o is unknown, we can estimate it from the observed trajectories.

The algorithm for reward advancement via Monte Carlo Policy Evaluation is summarized in Algorithm 1. Specifically, Line 3 calculates $Q_o^{\pi_t}(s,a)$ using Monte Carlo policy evaluation.

V. Evaluation

In this section, we first evaluate the correctness and accuracy of our (min-cost) reward advancement algorithm, with synthetic object world scenario. Then, by modeling passengers' travel decisions in public transit system as a Markov Decision Process, we conduct empirical case studies using a large-scale (6 months) passenger-level public transit data collected in Shenzhen, China, from 07/01/2016 to 12/30/2016.

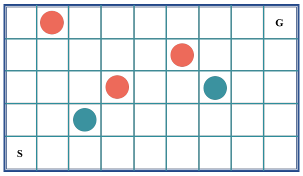


Fig. 2. A 5 × 9 Object World with 2 different colors.

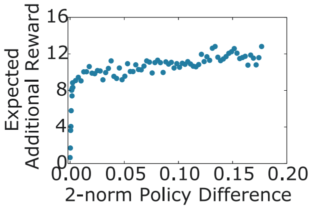


Fig. 3. Expected additional reward over policy difference.

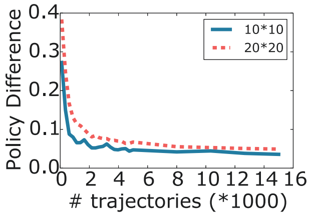


Fig. 4. Policy difference over number of trajectories used.

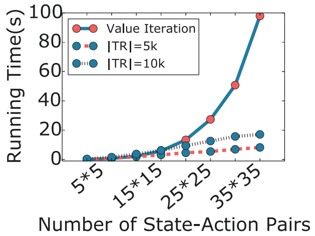


Fig. 5. Running time over the size of state space.

A. Evaluation on object world

First, we use an object world [18] scenario to evaluate our reward advancement algorithm. A Object World is a Grid World with random placed colored objects. Running into grids with objects in different colors will lead to different rewards. We call it "collect the object". The agent will also get a large reward by arriving the destination. So, the ideal policy should be going to the destination, while collecting as many objects with higher rewards as possible. Figure 2 shows an example of object world. There are 5×9 grids. We randomly placed 2 green objects and 3 red objects in the scenario. Each object has an color. An agent will gain a positive reward ($[5, 8]$ in our setting), when it reaches a grid with a red object, and a negative reward (ranging within $[-5, -3]$) when visiting a green object. A grid with no object leads to a negative reward of -1. Moreover, when the agent reaches the destination, it gets a large reward, within $[15, 30]$. At each grid, an agent can take 5 different actions, including "stay" and "move" towards one of four directions. With certain given transition probability, the agent would go to a random neighboring grid along the direction it has chosen. We choose discount factor γ to be 1 for all experiments. To evaluate our algorithm, we first randomly generate an Object World with pre-defined parameters, including the number of colors and objects. Then, we randomly place all objects in grids. We run value iteration with entropy-enhanced reward [19] to calculate a randomized original policy for the agent in the generated Object World. The target policy is also randomly generated as the objectives of reward advancement.

Impact of difference between π_o and π_t. First, we examine the impact of the difference between original policy and target policy to the expected additional reward. We use normalized 2-norm difference of two policy vectors to indicates the policy difference, which is calculated by $\|\pi_o, \pi_t\|_2 = \frac{\sum_{s\in S}\sum_{a\in A}(\pi_o(a|s)-\pi_t(a|s))^2}{|(s,a)|}$, where $|(s, a)|$ is the number of state-action pairs. We evaluate the expected additional reward, which is calculated as $\sum_{s\in S}\sum_{a\in A} D_t(s, a)\Delta R(s, a)$ indicating the total amount of additional reward we would provide. It's hard to design target policy with specific difference from original policy, so we group target policy with similar difference together and calculate the average ΔR. The result is shown in Figure 3. The figure indicates that with increase of policy difference, the expected additional reward would increase. When larger than a certain threshold of policy difference, the expected reward increases linearly, while the policy difference increases linearly as well. This shows that even with target policy far from original difference, we can successfully transfer the policy at a reasonable cost.

Impact of the number of trajectories used in Monte Carlo policy evaluation. Monte Carlo algorithm is employed to reduce algorithm running time. However, lower running time lowers down the inference accuracy. The smaller the sample size is, the less accurate the policy transformation is. Denote π'_t as the policy transformed to after executing Algorithm 1. Figure 4 shows how Monte Carlo sample size impacts the difference between π_t and π'_t. It is clear that more sampled trajectories lead to more accurate results. Roughly, we need around $5,000$ trajectories to achieve a relatively good accuracy. Moreover, with the increase of Object World size, more trajectories are needed to enable accurate estimation of additional rewards, for policy transformation.

Impact of state space size on algorithm running time. Though we have obtained the close-form solution of $\Delta Q(s, a)$ and $\Delta R(s, a)$, computing $Q_o^{\pi_t}(s, a)$ is still the bottleneck component, in running time. Here we evaluate the efficiency of the Monte Carlo based algorithm we proposed. Figure 5 shows the running time of 3 different algorithms. The blue solid line with red dot indicates running time of Value Iteration, the brown and red dash line with blue dot means running time of Monte Carlo method with $5,000$ and $10,000$ sampled trajectories, respectively. Obviously, the Monte Carlo method takes much less time than standard Value Iteration. Moreover, the running time of Monte Carlo method increases linearly, as the number of state increases quadratically. The running time increases with more trajectories used. The result shows that the Monte Carlo method is more computationally efficient.

B. Case studies

In this section, we use a real-world dataset to demonstrate the effectiveness of proposed reward advancement algorithm. To validate the algorithm, we need to verify that agent's behaviors still follow MCE principle after reward advancement. To validate this assumption, we use a large public transit dataset. We collected 6 months passenger-level public transit data from Shenzhen, China, which allows us to evaluate the potential of redistributing passengers by transforming their decision policies, in trip starting time, station and transport mode selection.

Passengers are making sequences of decisions when completing trips, such as which bus routes and subway lines to

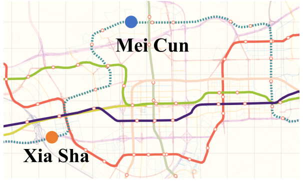


Fig. 6. Map with source and destination of one agent and the newly established subways.

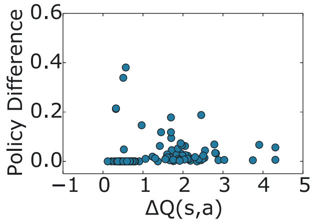


Fig. 7. Policy difference vs $\Delta Q(s, a)$.

take, which stops/stations to transfer at. Such sequential decision making processes can be naturally modeled as Markov decision processes (MDPs). Since nearby stops/stations usually are similar to passengers, we will split the whole city into grid cells and aggregate stops/stations within each grid cell. The states are regional grids during different time intervals. Actions are available bus routes and subway lines passengers can take. Our model and formulation follow the work [6]. We inversely learn the reward functions of passengers using Maximum Causal Entropy Inverse Reinforcement Learning [3] and the features we use include monetary cost, travel time, waiting time, and etc.

Additional reward can be provided using different methods, for example, deploying one new subway line can surely provide additional rewards to passengers in many different ways, for example, more transit choices, and lower average travel time. The Figure 6 illustrates the deployment of a new subway line in Shenzhen, which is the light blue dash line. To validate our assumption that providing additional rewards can transform passengers' behaviors to a target policy, we first learn the reward function from passengers' trajectories before deployment of the subway line. Then, we use features changed by deploying the new subway to calculate how much additional reward were provided. Lastly, we calculate a new policy, π_t, after reward advancement and compare this policy with ground-truth policy, π_{true}, of passengers after the deployment of new subway line.

Figure 7 shows that the policy differences of state-action pairs between π_t and π_{true} are small, with the X-axis as the Q-function difference of each state-action pair before and after subway deployment and the Y-axis as the relative error of π_t and π_{true}. The small difference between π_t and π_{true} validates that our reward advancement theory in Theorem 1.

VI. Conclusion

In this work, we define and study a novel reward advancement problem, namely, finding the updating rewards to transform human agent's behavior to a predefined target policy π_t. We provide a close-form solution to this problem. The solution we found indicates that there exist infinite many such additional rewards, that can achieve the desired policy transformation. Moreover, we define and investigate a min-cost reward advancement problem, which aims to find the additional rewards that can transform the agent's policy to π_t, while minimizing the cost of the policy transformation. We solve this problem by developing an efficient algorithm. We demonstrated the correctness and accuracy of our reward advancement solution using both synthetic data and a large-scale (6 months) passenger-level public transit data from Shenzhen, China.

VII. Acknowledgement

Guojun Wu and Yanhua Li were partly supported in part by NSF CRII grants CNS-1657350 and CMMI-1831140, and a research grant from DiDi Chuxing Research.

References

[1] G. Wu, Y. Ding, Y. Li, J. Luo, F. Zhang, and J. Fu, "Data-driven inverse learning of passenger preferences in urban public transits," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 5068–5073.

[2] S. Tao, D. Rohde, and J. Corcoran, "Examining the spatial–temporal dynamics of bus passenger travel behaviour using smart card data and the flow-comap," *Journal of Transport Geography*, vol. 41, pp. 21–36, 2014.

[3] B. D. Ziebart, "Modeling purposeful adaptive behavior with the principle of maximum causal entropy (CMU PhD dissertation)," 2010.

[4] R. Wong, W. Szeto, and S. Wong, "A two-stage approach to modeling vacant taxi movements," *Transportation Research Part C: Emerging Technologies*, vol. 59, pp. 147–163, 2015.

[5] L. Zhang, "Agent-based behavioral model of spatial learning and route choice," in *Transportation Research Board 85th Annual Meeting*, 2006.

[6] G. Wu, Y. Li, J. Bao, Y. Zheng, J. Ye, and J. Luo, "Human-centric urban transit evaluation and planning," in *IEEE International Conference on Data Mining, 2018*, 2018.

[7] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *AAAI*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

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

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

[10] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *International Conference on Machine Learning (ICML)*, vol. 99, 1999, pp. 278–287.

[11] G. Konidaris, I. Scheidwasser, and A. Barto, "Transfer in reinforcement learning via shared features," *Journal of Machine Learning Research*, vol. 13, no. May, pp. 1333–1371, 2012.

[12] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International Conference on Machine Learning*, 2016, pp. 49–58.

[13] M. Pan, Y. Li, X. Zhou, Z. Liu, R. Song, H. Lu, 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, pp. 783–791.

[14] 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)*. IEEE, 2018, pp. 547–556.

[15] X. Zhang, Y. Li, X. Zhou, and J. Luo, "Unveiling taxi drivers' strategies via cgail – conditional generative adversarial imitation learning," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 1–6.

[16] E. Wiewiora, "Potential-based shaping and q-value initialization are equivalent," *Journal of Artificial Intelligence Research*, vol. 19, pp. 205–208, 2003.

[17] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.

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

[19] J. Schulman, X. Chen, and P. Abbeel, "Equivalence between policy gradients and soft q-learning," *arXiv preprint arXiv:1704.06440*, 2017.