Inverse Reinforcement Learning for Legibility Automation in Intelligent Agents

Buxin Zeng, Yifeng Zeng
Department of Computer and Information Sciences
Northumbria University
Newcastle, UK
{buxin.zeng;yifeng.zeng}@northumbria.ac.uk

Yinghui Pan
National Engineering Laboratory for Big Data
Shenzhen University
Shenzhen, China
panyinghui@szu.edu.cn

***Abstract*—When intelligent agents operate in a stochastic environment, they adhere to the principle of maximizing expected rewards to optimize their policies. The maximization of rewards becomes the sole objective when agents' decision problems are resolved in most cases. However, there are instances where this principle leads to the agent's behaviors (the optimal policy for solving the decision problems) lacking *legibility*. In other words, comprehending the agents' intentions while they execute optimal policies poses a challenge for users, including other agents and even humans. Therefore, it becomes essential to evaluate the legibility of agents' decision-making processes. Traditionally, domain experts' insights have been relied upon to define legibility values, but this manual approach often introduces subjectivity and inconsistency, particularly in complex problem domains. Consequently, there is a pressing need for a systematic approach to derive legibility functions. The present study employs inverse reinforcement learning techniques to automate a legibility function in agents' decision problems. We demonstrate the effectiveness of the inverse reinforcement learning method when considering legibility in a decision problem. We vary problem domains in the performance study and provide empirical evidence to support our findings.**

***Index Terms*—Legibility, Inverse Reinforcement Learning, Decision Making, Intelligent Agents**

I. Introduction

Intelligent agent makes decisions to determine their actions to interact with the environment based on the rational principle in many AI applications. The rational principle usually means that the agents expect to maximize their total rewards. However, this decision-making process sometimes looks like lacking legibility. From a human perspective, it is challenging to figure out what their collaborators are doing [1]. For example, in a game where humans and multiple agents collaborate, such as Dota. The agent may not be able to communicate its intentions to a human when deciding to act, which may lead to the failure of the game. If the agents in the game can make their actions more legible, it will help human players understand their intentions. Thus, helping them better complete tasks that require collaboration.

Legibility has been a high concern in AI research in recent years. Dragan *et al.* [2] introduced the notion of legibility, that the observer can calculate the legibility value when observing the robot's path and infer the robot's goal. Capelli *et al.* [3] studied the legibility of a group of mobile robots. They found that trajectories are relevant to convey the robot's intention. Miura *et al.* [4] developed a method that maximizes the legibility in a stochastic environment to improve the interpretability of agent behavior. Liu *et al.* [5] proposed a reward-shaping method to improve the legibility of reinforcement learning. Their method can be further improved by interacting with the users. Most of the current research shows that from the human perspective, legibility can essentially help them understand the agent's intention. The research also shows that humans play the main role in calculating legibility. This implies that there is a legibility function that exists in reinforcement learning.

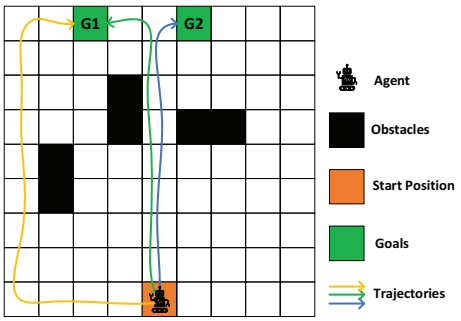
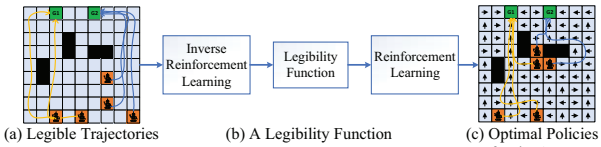


Fig. 1. Legible paths in a 9×9 Grid-World environment.

However, it is challenging to know the value of the legibility function. The existing research usually calculates the legibility function under some specific rules. A natural way to set the legibility value is to ask the domain experts to input them in the reinforcement learning environment. This is easy to assign in a simple domain. But when the domain becomes more complex, it seems impossible to assign specific values for the legibility function. Rather than assign specific values, it is more feasible for the domain experts to provide legible behaviors in a problem domain. For example, a set of legible paths is provided in Figure 1. If the agent starts from the middle at the bottom row and follows either the green or blue trajectory, the human observer can not predict which goal the agent expects to reach until it turns right or left in the top row. However, if the agent follows the yellow trajectory, it is clear that it is going to the goal G_1.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00143



(a) Legible Trajectories (b) A Legibility Function (c) Optimal Policies for the Agent

Fig. 2. We automate the development of the legibility function using the inverse reinforcement learning algorithm given the agent's legible trajectories from domain experts.

We will apply an inverse reinforcement learning [6] approach to learn the legibility function that the agent can use to learn the optimal legible policy in this paper. The legible trajectories shown in Figure 2 (a) are generated based on a set of navigation rules. Once we obtain the legible trajectories, we can adapt the inverse reinforcement learning to learn the legibility function (as shown in Figure 2 (b)). After the legibility function is learned, we perform the common reinforcement learning approach, such as Q-learning [7], to optimize the legible policy for the agent (as shown in Figure 2 (c)). We illustrate the learning process in the scalable Grid-World environment and show the empirical results of comparing the optimal policies.

The structure of this paper is as follows: Section II gives the background knowledge of reinforcement learning and inverse reinforcement learning. Section III shows how to learn legibility through the inverse reinforcement learning algorithm. Section IV reports the experiment results. We review the recent research about legibility in Section V. Section VI gives the conclusion on this work and some discussion for future study.

II. Background Knowledge

We briefly introduce the background knowledge of reinforcement learning and inverse reinforcement learning.

A. Reinforcement Learning

Intelligent agents determine their actions by maximizing the total discounted reward over a sequential of steps in reinforcement learning (RL) [8]. The decision-making process is usually modelled as a Markov decision process (MDP) and formulated as a tuple with five elements $M = \langle S, A, T, R, \gamma \rangle$, where S is a set of states, A is a set of actions, $T : S \times A \times S \rightarrow P(s_{t+1} \mid s_t, a_t)$ is the transition function denoting the probability of getting to the new state s_{t+1} when the agent takes the action a_t at the state s_t, R is a function of reward values that the agent receives when it takes the action a_t at the state s_t, and γ is the discount factor to dilute the future rewards. The learning process in reinforcement learning typically involves the agent's exploration of the environment, acquisition of knowledge from the consequences of its actions, and adaptation of its strategy to maximize the expected cumulative reward. The agent can acquire a policy that maps states to actions, and this policy is progressively refined through iterative learning. The Q-learning algorithm [7] is a technique that aims to learn a policy, represented by the Q-function, which estimates the expected cumulative reward associated with taking a specific action in a given state.

B. Inverse Reinforcement Learning

Inverse reinforcement learning (IRL) is a problem of finding/recovering a reward function R that best satisfies the given optimal policy π in a finite MDP [6]. One simple and natural optimization approach is to choose the function R that maximizes the sum of the deviation between the best action a_{opt} and the next-best action $a \in \{A \setminus a_{opt}\}$ over all states, as shown in (1). The general process of IRL is first to model the given trajectories (or policies) as solutions to MDP. Then, we use any given features to initialize the reward function R, and learn optimal behaviors (or policy) by solving the MDP using R. We repeatedly update R to minimize the deviation between the given trajectories and the learned behaviors until the deviation converges to an acceptable level. The elaborated detailed survey of inverse reinforcement learning can be seen in [9].

$$\sum_{s \in S} Q^{\pi}(s, a_{opt}) \; - \; max_{a \in \{A \setminus a_{opt}\}} Q^{\pi}(s, a) \qquad (1)$$

where Q is the Q-function value and a_{opt} is the optimal action.

III. Learning the Legibility Function

In this section, we use the IRL method to learn the legibility function. When a set of legible trajectories $\mathcal{D}_l = \{< (s_0, a_0), (s_1, a_1), \cdots, (s_t, a_t) >_1, \cdots, < (s_0, a_0), (s_1, a_1), \cdots, (s_t, a_t) >_n\}$ is given, where $s_t \in S$, $a_t \in A$, t is the decision time-step and n is the number of trajectories. We aim to learn the legibility function L that best satisfies $\mathcal{D}_l$ (if the trajectories are complete and optimal according to the corresponding MDP model, they are equal to the optimal legible policy π_l). We can ask the domain experts to provide the trajectory set or extract them from the displayed trajectories in the problem domain. We elaborate on the IRL process to learn a legibility function through the Grid-World domain.

A 9×9 Grid-World is shown in Figure 3 that has two goals G_1 and G_2 for the agent who needs to avoid the three obstacles (black grids) when it is moving. In the MDP setting, the agent can take three actions (Forward, Left and Right). It has the 80% chance of moving towards its expected direction while the remaining 20% is averaged to enter into both left and right directions. The agent will stay in the current grid when it moves towards a wall or obstacle. Based on a set of rules below, a set of legible trajectories are provided (towards G_1 and G_2 respectively). Some clearly legible samples are shown and start from the initial positions in the orange grids.

- We know the setting of the MDP except the legibility function. The legibility value is general for any of the goals and regardless of the start position of the agent.
- In the vertical direction, the agent's action is not legible whether the agent moves away from or to the goal.

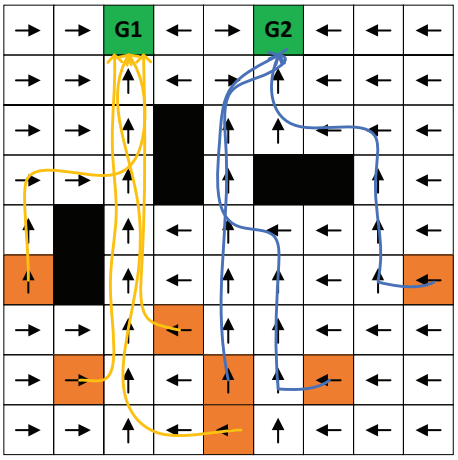


Fig. 3. The legible policy in the 9×9 Grid-World that provided from the domain experts. The arrows in the grid represent the best action the agent should take when it is at that state. The yellow and blue curves are representative of trajectories.

- In the horizontal direction, the agent's action is legible when it is clearly moving towards a goal; otherwise, it is not.
- The action moving towards a wall or obstacle is not legible.

We model the legible trajectories shown in Figure 3 as the optimal policy π_L as a solution to the MDP. Following the IRL process [6], we formulate the legibility function learning as the optimization problem in (2) subject to the constraints in (3) and (4). We use the Bellman optimality [10], [11] to derive (3). Obviously, we can use a linear programming method to find the solution to satisfy the constraints in (3). However, the majority of solutions, such as L=0 or any constant vector, are considered degenerate. Additionally, there remains a significant number of solutions beyond the degenerate ones, complicating the selection of the optimal solution. To address these challenges, we introduce the concept of a legibility function, denoted as L, which penalizes any deviation between the optimal policy π_L and the second-best policy. The goal is to design L in a way that maximizes the cumulative deviation between π_L and the second-best policy at each step, while still satisfying (4). Moreover, in the IRL framework, simplicity is favored in selecting the legibility function, provided it adheres to π_L. Hence, the IRL approach adds a weight decay-like penalty term $-\lambda\|L\|_1$, where λ is an adjustable penalty coefficient to formulate the IRL optimization as (2). By applying well-established optimization techniques, we derive the optimal legibility function through solving this problem.

$$maximize \quad \sum_{i=1}^{N} min_{a\in\{A\setminus a_{opt}\}}\{(T_{a_{opt}}(i) - T_a(i)) (I - \gamma T_{a_{opt}})^{-1}L\} - \lambda\|L\|_1 \tag{2}$$

$$s.t. \quad (T_{a_{opt}} - T_a)(I - \gamma T_{a_{opt}})^{-1}L \succeq 0 \quad \forall a \in A \setminus a_{opt} \tag{3}$$

$$|L_i| \leq L_{max}, i = 1, \cdots, N \tag{4}$$

where a_{opt} is the optimal action in the policy π_L, $\{A \setminus a_{opt}\}$ are the other actions in the action set A except a_{opt}, $T_a(i)$ is the ith row of the transition probability matrix T, I is the identity matrix, γ is the discount factor in the MDP, L is the legibility function, L_{max} is the maximum legibility value and N is the number of the states.

IV. Experimental Results

We executed experiments in the scaled Grid-World problem domains to illustrate the performance of the method that uses IRL to learn the legibility function. We conducted the experiments in both the 9×9 and 13×13 settings of the Grid-World. We compare the performance of four types of the agent's policies learned in these two domain settings: (a) IP: the policy that is given from the domain experts; (b) LP: the policy learned from the legibility function that is extracted from the full-size input trajectories through the IRL method; (c) LP($|\mathcal{D}|$=1000) and (d) LP($|\mathcal{D}|$=500): the policy learned from the legibility function through IRL given a subset of $\mathcal{D}$ where the number of the trajectories is equal to 1000 and 500. All the implementations were conducted in Windows 10 with the setting of CPU (11^{th} Gen Intel Core i7-11800H @ 2.30GHz 8-core) and 32 GB RAM.

A. The 9×9 Grid-World Environment

Given the set of trajectories that are shown in Figure 3, $\mathcal{D}_l$, we model them as an optimal policy π_L for the 9×9 Grid-World domain. Then we solve the optimization problem in (2) subject to the constraints in (3) and (4). The result is shown in Figure 4. The colour of the grids represents the value between 0 to 1. The darker the colour, the smaller the legibility value. Figure 4 shows that the legibility value of the states is clearly higher around the two goals than any other states.

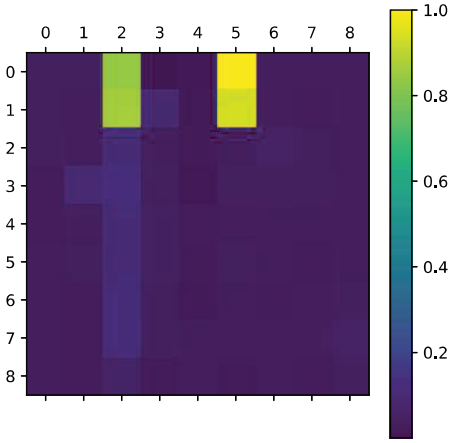


Fig. 4. The legibility function learned from the legible trajectories in Figure 3. The colour represents the value distribution between 0 to 1.

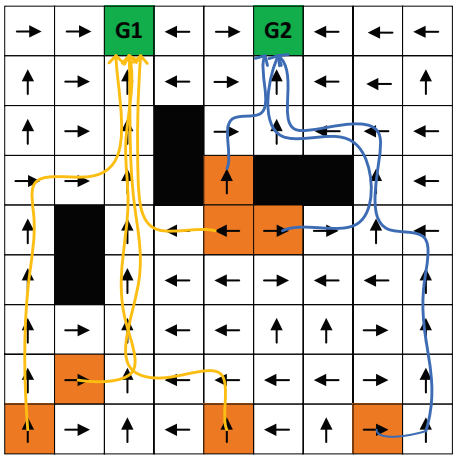


Fig. 5. The legible optimal policy learned using the learned legibility function in Figure 4.

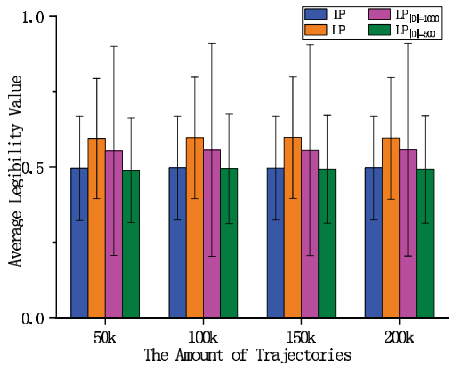


Fig. 6. The average reward values are reported for different policy inputs when the agent navigates in the 9×9 Grid-World environment. The sample number of the trajectories is 50000, 100000, 150000 and 200000.

Hence, we use Q-learning to learn the legible optimal policy using the legibility function in Figure 4 and the optimal policy is shown in Figure 5. The learned policy is not completely the same as the given trajectories in Figure 3. However, the column 5 of Figure 5 shows more clear and intensive navigation paths that the agent will go to G_1. The policy also shows that the agent tends to determine where it wants to go first.

To evaluate the performance of the learned policy, let the agent walk in the Grid-World environment based on the different optimal policies. The agent starts from a random start position and stops at any of the goals. The agent will receive a reward when it takes one action until it reaches any of the goals. The sum of the rewards is called the reward of a trajectory (path). We calculate the average of the trajectory reward when the number of the trajectories is 50000 to 200000 for each optimal policy. The experiment result of the average reward is shown in Figure 6.

The agent receives a higher reward value when it performs

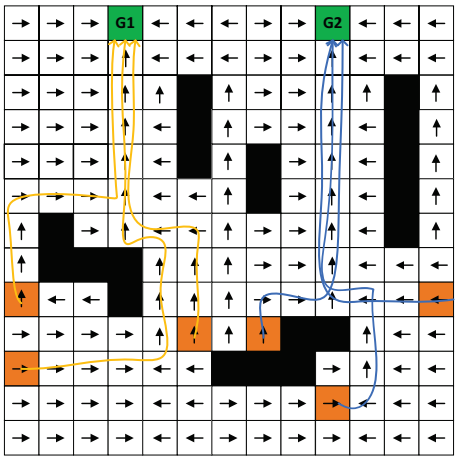


Fig. 7. The legible policy in the 13×13 Grid-World that provided from the domain experts. The arrows in the grid represent the best action the agent should take when it is at that state. The yellow and blue curves are representative of trajectories.

based on the learned policies LP than the input policies IP. It illustrates that the IRL method performs well in learning the legibility function. Especially we examine the performance of the learned policies when a subset of the trajectories $\mathcal{D}_l$ are given into the algorithm. The results show that the IRL algorithm does not perform as well as the situation in which the full trajectories are given. This is expected as a subset of trajectories can not provide sufficient information to find optimal policies in the learning process. Hence, the inverse reinforcement learning algorithm could learn the legibility function from the agent's legible behaviors. And it expresses better policies in some cases.

B. The 13×13 Grid-World Environment

In this section, we test the inverse reinforcement learning algorithm in learning the legibility function in a more complex problem domain. The 13×13 Grid-World environment has more than twice the states than the 9×9 one. More black obstacles are added to the environment. The same experimental setting in Section IV-A is adopted in this experiment.

The given legible trajectories from the domain experts are shown in Figure 7. We execute the inverse reinforcement learning algorithm using the legible trajectories and the learned legibility function is shown in Figure 8. The legibility function also shows that the states around the two goals have a higher legibility value as exhibited in the 9×9 domain. Hence the IRL still performs well in learning the legibility function in the complex domain. Q-learning is used to learn the legible optimal policy for the agent using the learned legibility function and show the policy in Figure 9. Some typical legibility trajectories are provided in Figure 9. It is expected that the optimal policy is not completely the same as the input trajectories in Figure 7 but shows the agent's intention that it wants to first determine its purpose.

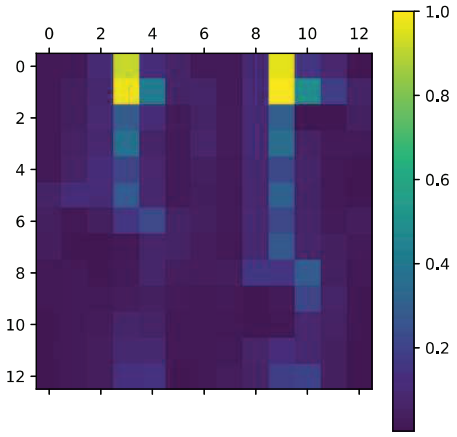


Fig. 8. The legibility function learned from the legible trajectories in Figure 7. The colour represents the value distribution between 0 to 1.

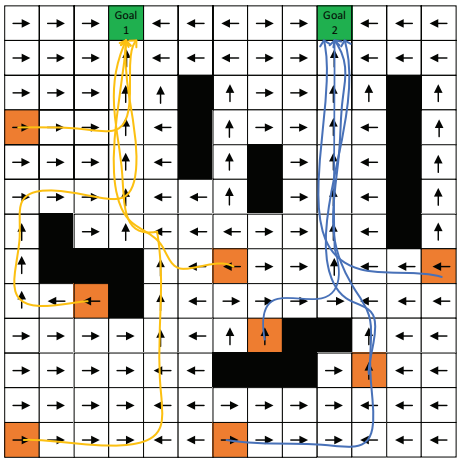


Fig. 9. The legible optimal policy learned using the learned legibility function in Figure 8.

The mean legibility value obtained by the agent while operating in the Grid-World environment is shown in Figure 10, based on distinct optimal policies derived from 4 sets of trajectories. The experiment results show that the average legibility value is higher when given the full legible behaviors than in any other cases. The results are similar to the 9×9 Grid-World. The computational times for the IRL algorithm in learning the legibility function within the 9×9 and 13×13 Grid-World domains are presented in Table I. The running times exhibit minimal variation when the state space remains constant, as the majority of computational effort is devoted to executing the IRL algorithm. The slight discrepancy in computational time for the same state space may be attributed to the relatively limited duration allocated for processing distinct sets of input trajectories. The expansion of the state space will result in an exponential growth in the computational demands imposed by the IRL algorithm.

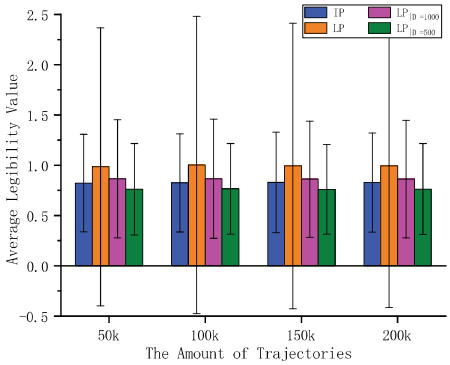


Fig. 10. The average reward values are reported for different policy inputs when the agent navigates in the 13×13 Grid-World environment. The sample number of the trajectories is 50000, 100000, 150000 and 200000.

TABLE I
RUNNING TIMES (SECONDS) FOR LEARNING THE LEGIBILITY FUNCTION

State Space	*Method*	*Running Times*
9×9	LP($\|\mathcal{D}\|$=500)	30.55
	LP($\|\mathcal{D}\|$=1000)	31.34
	LP	31.63
13×13	LP($\|\mathcal{D}\|$=500)	2651.9
	LP($\|\mathcal{D}\|$=1000)	2652.13
	LP	2653.87

V. Related Works

In light of the dynamic progress in AI technology and its expanding integration into real-world contexts, there is a growing imperative for humans to deepen their understanding of AI legibility [12]. This section is dedicated to a comprehensive review of relevant research on legibility.

The research on legibility in AI has some different applications. The most relevant research is on agents' legibility in multi-agent systems. Miura and Zilberstein [13] proposed a method to balance the trade-off between maximizing the legibility and the underlying reward in the MDP. They demonstrated that maximizing legibility results in legible behaviors. Miura *et al.* [4] developed legible MDPs in which the agent aims to convey its intentions clear to the observer. They proved that maximizing legibility results in more interpretable behaviors. Mavrogiannis *et al.* [14] presented a planning framework that enabled agents to communicate their intentions to avoid collision with other agents in multi-agent environments. Their framework consistently achieved significantly lower topological complexity in multi-agent collision avoidance. Kulkarni *et al.* [15] presented a planning framework to simultaneously convey legible information to collaborative agents and obfuscate information to adversarial agents based on the assumption that the different observers have differing sensing capabilities. Habibian and Losey [16] introduced an optimization approach that enables robot teams to optimize for legibility and fairness. They showed that humans prefer to participate and collaborate with a legible agent team which is not just concerned with

efficiency. Liu *et al.* [5] exhibited the benefit of legibility to agents' decision-making; however, it was still challenging to develop a rational metric to evaluate the legibility. Bied and Chetouani [17] developed an approach to integrate the observer's feedback into a reinforcement learning framework to improve the legibility of the robot's trajectories.

It is valuable for human beings to understand robots' intentions since they are widely used in industry. Dragan and Srinivasa [18] proposed a functional gradient optimization technique for autonomously generating legible motion to deviate the robot's trajectories from the observer's expectation to better convey the robot's intentions. Wallkotter *et al.* [19] showed that robots' movement trajectories could communicate their intentions and they provided an approach to use the trajectories which are independent of the tested ones to evaluate the legibility. Nikolaidis *et al.* [20] proposed viewpoint and occlusion models that enable autonomous generation of viewpoint-based legible motions and showed that when the robot's trajectory can provide more information about its goals, the observer can infer its intention more quickly and confidently.

To the extent that legibility is a human subjective attribute, some studies have considered these. The view of observers is vital for the legibility, Taylor *et al.* [21] designed an observer-aware method for creating navigation paths that concern the view of observers to make the paths legible. Hetherington *et al.* [22] demonstrated that robots' legibility cues are helpful for humans to understand the robots' behaviors.

The typical approach in current research assumes a known legibility function for solving policy optimization problems. However, evaluating legibility still depends on substantial input from domain experts. It becomes important to research the legibility of auto-learning in multi-agent systems.

VI. Conclusion and discussion

We propose an IRL-based method to learn the legibility function in this paper. The approach can estimate the legibility function from the input of the agent's legible behaviors. We elaborate on the learning process through the Grid-World example and demonstrate the learning performance by varying the problem set. The experiment shows that the learning accuracy will decrease when the number of input legibility trajectories becomes smaller. Future research is to focus on selecting a small set of trajectories while retaining sufficient legible information. The new legibility research also contributes to solving complex multiagent decision-making problems. We are considering the integration of the legibility function in a general decision-making framework, namely interactive dynamic influence diagrams [23], in multiagent systems.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (Grants No.62176225 and 62276168) and the Natural Science Foundation of Guangdong Province, China(Grant No. 2023A1515010869).

References

[1] "Can bounded and self-interested agents be teammates? application to planning in ad hoc teams," *Autonomous Agents and Multi-Agent Systems*, vol. 31, pp. 821–860, 2017.

[2] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 301–308, IEEE, 2013.

[3] B. Capelli, V. Villani, C. Secchi, and L. Sabattini, "Understanding multi-robot systems: on the concept of legibility," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7355–7361, IEEE, 2019.

[4] S. Miura, A. L. Cohen, and S. Zilberstein, "Maximizing legibility in stochastic environments," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pp. 1053–1059, IEEE, 2021.

[5] Y. Liu, Y. Zeng, B. Ma, Y. Pan, H. Gao, and X. Huang, "Improvement and evaluation of the policy legibility in reinforcement learning," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 3044–3046, 2023.

[6] A. Y. Ng, S. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, p. 2, 2000.

[7] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[8] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[9] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

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

[11] D. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[12] Y. Zeng, H. Mao, Y. Pan, and J. Luo, "Improved use of partial policies for identifying behavioral equivalence," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, p. 1015–1022, 2012.

[13] S. Miura and S. Zilberstein, "Maximizing plan legibility in stochastic environments," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1931–1933, 2020.

[14] C. I. Mavrogiannis, W. B. Thomason, and R. A. Knepper, "Social momentum: A framework for legible navigation in dynamic multi-agent environments," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 361–369, 2018.

[15] A. Kulkarni, S. Srivastava, and S. Kambhampati, "Signaling friends and head-faking enemies simultaneously: Balancing goal obfuscation and goal legibility," *arXiv preprint arXiv:1905.10672*, 2019.

[16] S. Habibian and D. P. Losey, "Encouraging human interaction with robot teams: Legible and fair subtask allocations," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6685–6692, 2022.

[17] M. Bied and M. Chetouani, "Integrating an observer in interactive reinforcement learning to learn legible trajectories," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 760–767, IEEE, 2020.

[18] A. Dragan and S. Srinivasa, "Generating legible motion," 06 2013.

[19] S. Wallkotter, M. Chetouani, and G. Castellano, "A new approach to evaluating legibility: Comparing legibility frameworks using framework-independent robot motion trajectories," *arXiv preprint arXiv:2201.05765*, 2022.

[20] S. Nikolaidis, A. Dragan, and S. Srinivasa, "Viewpoint-based legibility optimization," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 271–278, IEEE, 2016.

[21] A. V. Taylor, E. Mamantov, and H. Admoni, "Observer-aware legibility for social navigation," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1115–1122, IEEE, 2022.

[22] N. J. Hetherington, E. A. Croft, and H. M. Van der Loos, "Hey robot, which way are you going? nonverbal motion legibility cues for human-robot spatial interaction," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5010–5015, 2021.

[23] P. Doshi, Y. Zeng, and Q. Chen, "Graphical models for online solutions to interactive pomdps," in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.