A study of inverse reinforcement learning and its implementation

Chaoying Zhang*[a], Guanjun Jing[a], Siqi Zuo[b], ZhiMing Dong[a]
[a]Beijing Research Institute, China Telecom Corporation Limited, Beijing 102209, China;
[b] Beijing Institute of Technology, Beijing, 10008, China.
[*] Corresponding author: zhangchy50@chinatelecom.cn

ABSTRACT

When dealing with complex tasks, such as robots imitating human actions and autonomous vehicles driving in urban environments, it can be difficult to determine the reward function of the Markov decision-making process. In contrast to reinforcement learning, inverse reinforcement learning (IRL) can infer the reward function through the finite state space and the linear combination of reward features, given the optimal strategy or expert trajectory. At present, IRL has many challenges, such as ambiguity, large computation and generalization. As part of this paper, we discuss existing research related to these issues, describe the existing traditional IRL methods, implement the model, and then propose future direction for further research.

Keywords: Inverse Reinforcement Learning, Reward Function, Expert Trajectory, Maximum Entropy Optimization

1. INTRODUCTION

In recent years, reinforcement learning (RL) has played a great role in artificial intelligence, such as defeating the world's top Go players through alpha go, and robots imitating and learning human actions and expressions. Reinforcement learning is to update the strategy according to the known reward function through the interaction between the agent and the environment, so as to promote the agent to make the best action and maximize the reward [1]. However, reinforcement learning is no longer applicable to complex scenarios where the reward function is difficult to determine. For example, the task of driving a vehicle from the starting point to the destination in a complex urban environment needs to consider factors such as traffic rules, passenger comfort and efficiency of reaching the destination, which is difficult to build a suitable reward function [2]. However, an experienced driver can fully consider the above problems. Therefore, we can learn the reactions of drivers in different environments during driving to help machines understand the reward. Contrary to the process of reinforcement learning, inverse reinforcement learning (IRL) is to find the reward function that can explain these strategies with the optimal strategies or expert demonstrations. The RL and IRL models are shown in Figure 1, where dynamic model represents the state distribution at the next moment corresponding to the action in the current state, optimal strategy represents the action taken in the current state that maximizes the reward function, and reward function refers to the reward mapping obtained by taking the action in the current state.

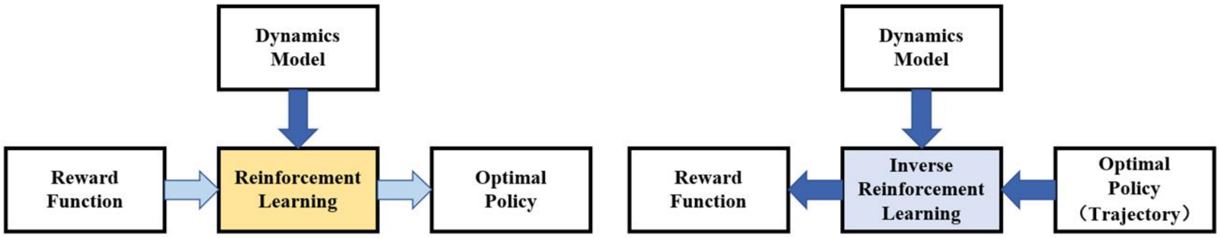
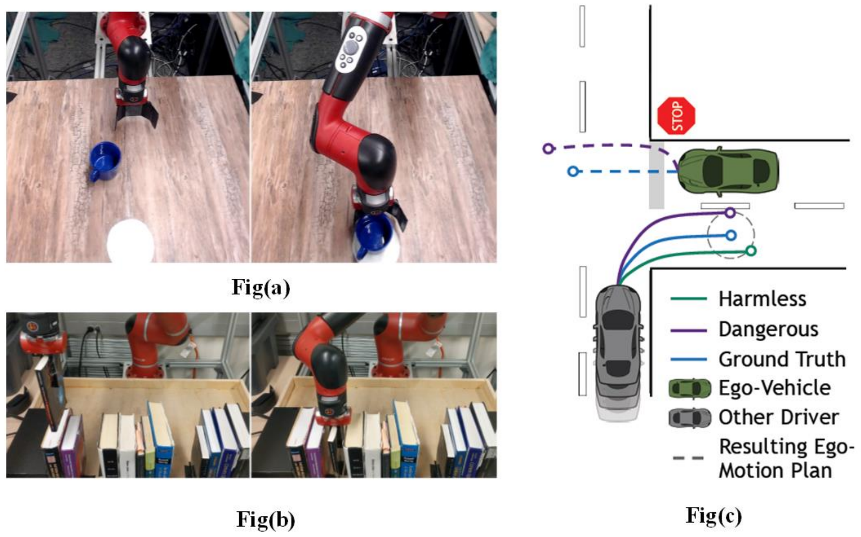


Figure 1. Reinforcement learning and inverse reinforcement learning model.

Inverse reinforcement learning, also known as inverse optimal control, has developed rapidly in the past decade with the upsurge of deep learning since it appeared 20 years ago. On the one hand, IRL can be applied to decision-making and planning areas to reduce manual adjustment in tasks. For example, for multi-objective optimization problems, the relative importance of each task can be determined through IRL. On the other hand, through IRL and RL, robots can imitate human actions or decisions. Imitation learning uses demonstration to infer hidden reward functions, and then uses inferenced reward functions in reinforcement learning to learn imitation strategies [3-4]. The application of inverse

Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022),
edited by Xianye Ben, Proc. of SPIE Vol. 12462, 124620E · © The Authors. Published under a
Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2661162

reinforcement learning is shown in Figure 2. Fig (a) shows that the robot arm pushes the cup onto the coaster, and the initial position of the cup is random. Fig (b) shows the robot arm insert a book into the empty slot in the bookshelf. Fig (c) shows planning task of automatic driving.



Fig(a)

Fig(b)

Fig(c)

Figure 2. Application of inverse reinforcement learning.

Inverse reinforcement learning is a new subject in the field of machine learning developed in recent years, so there are many challenges. First, because of the limited number of expert examples, the strategy will correspond to a variety of reward functions, and the ambiguity problem will hinder the accuracy of inference. Secondly, practical inverse reinforcement learning should show good generalization. The model needs to infer other unobserved learned state actions, and the application cannot be limited to specific scenarios, which challenges the correct extension of a small part of data to unobserved space. Third, considering that there are a large number of noise data in the environment, inaccurate prior information will enter the characteristic equation of IRL, so the model should be robust enough and ensure accuracy. Finally, IRL needs to consider the complexity of space and time. Due to the complexity of the task, the state and action space are generally high-dimensional vectors after discretization, and the computation is exponential with the state vector, which consumes much computing time and space [5].

This paper mainly introduces the current situation and challenges of inverse reinforcement learning. In the following sections: we list the significance and existing problems of IRL in section 1. Next, we introduce the basic principles and models of IRL in section 2. The third part mainly introduces two basic IRL methods. Then, we implement IRL model in a vehicle planning tasks in section 4. Finally, the paper puts forward possible future development trends in section 5.

2. BASIC PRINCIPLE

Markov decision process (MDP) is the model foundation of reinforcement learning, multi-task decision making, and interaction between agents and environment. By assuming Markov properties, MDP means that the current state is only related to the state at the last time and unrelated with the previous state. MDP is to find an optimal strategy to maximize the state reward function [6]. Therefore, we define S as a finite set of States, A as an action set. The policy represents the mapping function between the current state and the action at the next moment. It can be defined as $\pi: S \rightarrow A$. Reward function $R(s,a)$ is a scalar return obtained by taking actions a in the current state s. The Markov decision process model is shown in Figure 3.

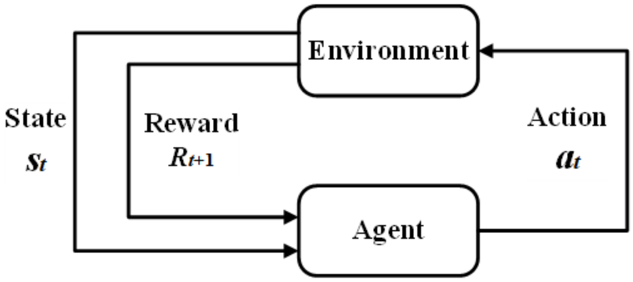


Figure 3. Markov decision process model.

Inverse reinforcement learning assumes that expert behavior is based on a certain strategy. If the strategy is unknown, it can be obtained by observing the state action pair of expert behavior. General reward functions are defined as linear combinations of reward features.

$$\begin{aligned} R(s,a) &= w_1\phi_1(s,a) + w_2\phi_2(s,a) + \ldots + w_k\phi_k(s,a) \\ &= \mathbf{w}^T\phi(s,a) \end{aligned} \tag{1}$$

These reward features ϕ_k is feature function about state s and action a, with the corresponding weight w_k.The IRL algorithm flow is shown in Table 1.

Table 1. IRL algorithm flow [7]

IRL algorithm flow:

Input: example trajectory or expert strategy, features of reward function;

Output: inferred expert reward function;

1. Modeling expert behavior as rewarding unknown MDP;

2. Initialize the parameters of the reward function (linear feature parameters or distribution of rewards to states);

3. Use the current reward function to solve the MDP and get the current strategy and behavior sequence (online algorithm);

4. Optimize parameters: minimize the difference between the sample behavior sequence (or strategy) and the current behavior sequence (or strategy);

5. Repeat step 3 and step 4: reduce the difference to the set value.

3. METHODS

The current mainstream IRL methods can be divided into margin-based optimization and entropy-based optimization through different reward functions. Among them, the milestone research breakthrough node is Ziebart's maximum entropy optimization method, which solves the ambiguity of IRL.

3.1 Margin optimization method

The purpose of the margin-based optimization method is to find a reward function that is better in the example strategy than in the learning strategy, that is, the margin between the two strategies is the largest [8]. When the margin decreases to the set value, convergence is considered. For any track τ, margin formula is as follows.

$$\sum_{\langle s,a\rangle\in\tau_i} \psi(s)\mathbf{w}^T\phi(s,a) - \max_{\tau\in(S\times A)^l\setminus\{\tau_i\}} \sum_{\langle s,a\rangle\in\tau} \psi(s)\mathbf{w}^T\phi(s,a) \tag{2}$$

Margin based optimization is to solve the ambiguity of IRL solution by convergence maximum margin.

Later, Ratliff proposed an improved method, learn to search (LEARCH) [9], so as to convert the quadratic programming problem into an optimization problem, and then solve it by solving Hessian matrix and gradient, so as to solve the difficult problem of solving high-dimensional continuous time problems.

3.2 Entropy optimization method

IRL method has the problem of ambiguity, which means more than one reward functions can explain the behavior of experts, so the maximum marginal method will introduce a random bias. In order to eliminate this bias, the maximum entropy optimization (MaxEnt) obtains the behavior distribution through the maximum entropy principle. Specifically, this method assumes that all trajectory distributions are probability distributions, and only needs to solve the probability model that generates the expert trajectory distribution [10]. Since the probability distribution of maximum entropy does not make any assumptions about any other unknown information except for constraints, this method can avoid ambiguity, and parameterized by the weight of the reward function.

MaxEnt method needs to solve the model with maximum entropy, which is essentially a nonlinear convex optimization problem. The maximum entropy of the policy distribution [11] is as follows.

$$\max_{D} - \sum_{\pi \in (S \times A)} \Pr(\pi) \log \Pr(\pi) \tag{3}$$

where D represents all distributed spaces. For probability distribution P and Q , the improved relative entropy model [12] is as follows.

$$\min_{P \in \Delta} \sum_{\tau \in (S \times A)^l} P(\tau) \log \frac{P(\tau)}{Q(\tau)}. \tag{4}$$

4. EXPERIMENT AND ANALISIS

In this paper, IRL based on maximum entropy optimization is used to modify the trajectory prediction index in unmanned missions. Traditional metrics, such as average or final displacement error (ADE / FDE), do not consider the follow-up planning tasks. Therefore, IRL is used for metric improvement, and the linear combination as shown in formula (1) is constructed by considering the factors of vehicle collision, control term and distance to the target point [13].

$$R\left(s^t, u_{\mathrm{R}}^t, \hat{s}^{t:T}\right) = w_1 \varphi_1\left(\left\|s_{\mathrm{ego}}^{(t)}\right\|_2^2\right) + w_2 \varphi_2\left(\left\|u^t\right\|_2^2\right) \quad + w_3 \varphi_3\left(\left\|s_{\mathrm{ego}}^t - s_a^t\right\|\right) + w_4 \varphi_4\left(\left\|\hat{s}_{\mathrm{ego}}^{t+1} - \hat{s}_a^{t+1}\right\|\right) \tag{5}$$

where s_{ego}^t is the position x, y of the ego-vehicle at time t and s_a is the obstacles' position; $\hat{s}_{\mathrm{ego}}^{t+1}$ is the predicted position of the ego-vehicle at time t + 1, the same applies to obstacles; u is a control variable, including heading and velocity control.

Build the experiment based on the gym-collision-avoidance simulation environment and use limited-memory BFGS (L-BFGS) to optimize. The time step of the model is 0.1s, and the number of time steps of each track is 50. By selecting the reward features of the above formula, the weights are obtained as follows in Table 2, where the slack variable is the minimum value that the optimizer can reach for convergence, φ_2^v means only use velocity to control and φ_2^{head} means only use heading to control.

Table 2. Different learned weight and variable according to reward features.

Number	Reward features	Learned weight	Slack variable
1	$\varphi_1, \varphi_2, \varphi_3, \varphi_4$	[1.214, 4.188, 0.366, 0.352]	1e-07
2	$\varphi_1, \varphi_2^v, \varphi_4$	[2.508, 4.607, 1.097]	0.18
3	$\varphi_1, \varphi_2^{head}, \varphi_4$	Not converge	None
4	φ_1, φ_4	[0.345, 0.085]	0.25

By training the first reward function, the convergence results of the model are as follows in Figure 4. The left figure is the change curve of slack variable. It can be seen that as the number of iterations increases, it gradually converges to approach the best point. The right figure shows the training loss of negative loglikelihood (NLL). The weight change curves of four reward features are as follows in Figure 5. After 2.0 iterations, the weight gradually converges to a fixed value.

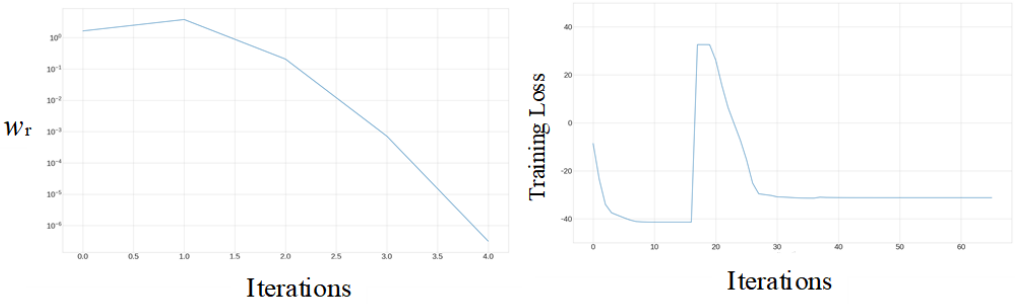


Figure 4. Convergence results of the model.

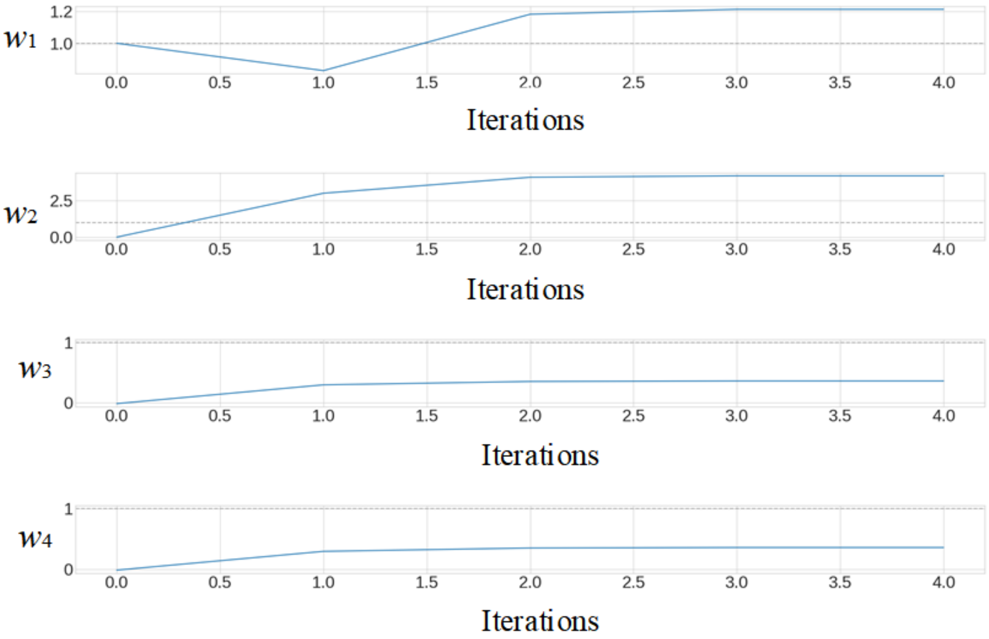


Figure 5. Weight change curve of four reward features.

5. SUMMARY AND FUTURE DIRECTION

Aiming at the fuzziness of IRL, the entropy optimization method models the uncertainty of reward as the probability distribution on the trajectory of reward. In addition to the above two methods, IRL algorithms also include models based on Bayesian update, which can be divided into Boltzmann distribution (BIRL), Gaussian process (GPIRL) [14] and maximum likelihood estimation according to different models of observation likelihood.

At present, scholars are exploring new solutions for practical application. When the trajectory is too long and the dimension is too large, considering suboptimal algorithm or local optimization becomes a solution. For example, in the task from city a to city B, mature drivers can complete it perfectly at every intersection, but it is still difficult to consider the overall optimization. In the future, we may study the direct and indirect learning algorithm and analyze the complexity and accuracy of the IRL algorithm in view of the high-dimensional number of states, so as to better apply to reality.

REFERENCES

[1] Ibarz J., Tan J. and Finn C, "How to train your robot with deep reinforcement learning: lessons we have learned," The International Journal of Robotics Research, 698-721 (2021).

[2] Bachute M. R. and Subhedar J. M. "Autonomous driving architectures: insights of machine learning and deep learning algorithms," Machine Learning with Applications, 6: 100164 (2021).

[3] Shani L., Zahavy T. and Mannor S., "Online apprenticeship learning," Proceedings of the AAAI Conference on Artificial Intelligence, 36(8): 8240-8248 (2022).

[4] Ravichandar H., "Recent advances in robot learning from demonstration," Annual review of control, robotics, and autonomous systems 297-330 (2020).

[5] Saurabh A. and Prashant D., “A survey of inverse reinforcement learning: Challenges, methods and progress,” Artificial Intelligence, 297 (2021).

[6] Zhou D., Gu Q. and Szepesvari C., "Nearly minimax optimal reinforcement learning for linear mixture markov decision processes," Conference on Learning Theory,4532-4576 (2021).

[7] Arora S. and Doshi P., "A survey of inverse reinforcement learning: Challenges, methods and progress," Artificial Intelligence, 297: 103500 (2021).

[8] Bashir E. and Luštrek M., "Inverse Reinforcement Learning Through Max-Margin Algorithm," Intelligent Environments 2021: Workshop Proceedings of the 17th International Conference on Intelligent Environments, 109 (2021).

[9] Bashir E. and Luštrek M., "Inverse Reinforcement Learning Through Max-Margin Algorithm," Intelligent Environments 2021: Workshop Proceedings of the 17th International Conference on Intelligent Environments, 29: 190 (2021).

[10] Bogert K., Gui Y. and Doshi P., "IRL with Partial Observations using the Principle of Uncertain Maximum Entropy," arXiv preprint arXiv:2208.06988 (2022).

[11] Mehr N., Wang M. and Schwager M., "Maximum-Entropy Multi-Agent Dynamic Games: Forward and Inverse Solutions," arXiv preprint arXiv:2110.01027 (2021).

[12] Qu B., Zhao M. and Feng J., "ASRL: An Adaptive GPS Sampling Method Using Deep Reinforcement Learning," 2022 23rd IEEE International Conference on Mobile Data Management (MDM), 153-158 (2022).

[13] Boris I., and Pavone M., "Rethinking trajectory forecasting evaluation," arXiv preprint arXiv:2107.10297 (2021).

[14] Nasernejad P., Sayed T. and Alsaleh R., "Modeling pedestrian behavior in pedestrian-vehicle near misses: A continuous Gaussian Process Inverse Reinforcement Learning (GP-IRL) approach," Accident Analysis & Prevention, 161: 106355 (2021).