Survey of Algorithms for Inverse Reinforcement Learning

Shaun Pritchard * CAP6629 Reinforcement Learning * Florida Atlantic University * spritchard2021@fau.edu *

I. INTRODUCTION

Specifically, this paper considers inverse reinforcement learning in Markov decision processes with respect to the problem of extracting a reward function that gives the observed optimal behavior. Based on an agent's policy or observed behavior, inverse reinforcement learning attempts to infer its reward function. A problem of IRL is the modeling of another agent's preferences by observing its behavior, rather than defining its reward function manually. In the following paper, the authors first characterize a set of all reward functions for which an optimal policy can be derived, before deriving three for IRL. The goal of inverse reinforcement learning is to discover a true reward function that can explain observed behavior. The research proposes instead to recover the experts' reward function and to use this to generate desirable behavior by suggesting that the reward function provides more information about the behavior. IRL suggests that the task is to take a set of human or animal(sentient behavioral data) such as generated driving for example. The research explains extracting an approximation of that sentient actions to map interposable behaviors that can be computed into a reward function for the task. Still, solving the problem is captured within the approximation of the true reward function being the aim of this research. Once there is a correct reward function, the problem is reduced to finding the right policy, and can be solved with standard reinforcement learning methods. In the research, the key issue is the issue of degeneracy, the existence of a large set of rewards functions for which the observed policy excels. To remove degeneracy, the researchers proposed natural heuristics that attempt to pick out every board function that differs maximally from the observed policy. This study derived three algorithms for its test case, two which were demonstrated on discrete, continuous, finite, and infinite State problems. Two of the algorithms deals with the case where the model is known, and the complete policy is known.

The third algorithm characterizes the set of all reward functions for which a given policy is optimal and known. A key issue pertains to degenerate solutions including for example the reward function that is identical 30 everywhere they resolve this difference using heuristic attempts to identify a reward function that maximally differentiates between the observer policy and other sub-optimal policies using linear programming methods. The algorithm-based experimentation is defined in more realistic cases in which the policy is known only through a finite set of observed trajectories using simple iterative algorithms. The research also defined algorithms based on infinite state spaces for which an explicit tabular representation of the word funk but be invisible they show that if the fitted reward function is represented as a linear combination of arbitrary fixed bias function. Then the IRL problem remains in the class of linear programs and can again be solved efficiently. In this paper a survey was conducted based on Inverse Reinforcement Learning problem as classified in the research paper by Professor Andrew Y. Ng and Stuart Russell. Citations to several other papers are noted to give context to subject matter in relation to the survey topic. This paper is as follows: Introduction to the survey, algorithms and design scope to the experiments with the research paper, results and discussion section which details the topic and discuss the findings and approximations of the experimentations, and then the Conclusion section which derives the final results and open discussion of the finding and context in relation to the research paper [1].

II. ALGORITHMS

The experiments are based on the Markov decomposition process, Monte Carlo simulation, and the Q-learning algorithm with several examples and implementations to derive the value function and bellman function and bellman optimality. The policies for each extent are described as under the characteristic of the sections discussed in the original paper. As basic properties of MDPs, the researchers used two classical results pertaining to Markov

decision processes - the Bellman equation and Bellman optimality. The MDP is defined as follows

A (finite) MDP is a tuple $(S, A, \{P_{sa}\}, \gamma, R)$, where

- S is a finite set of N **states**.
- $A = \{a_1, \ldots, a_k\}$ is a set of k **actions**.
- $P_{sa}(\cdot)$ are the state **transition probabilities** upon taking action a in state s.
- $\gamma \in [0, 1)$ is the **discount factor**.
- $R : S \mapsto \mathbb{R}$ is the **reinforcement function**, bounded in absolute value by $R_{\max}$.

Rewards was defined as $R(s)$ as oppressed to $R(s, a)$ for simplicity. A **policy** function is defined as any map $\pi : S \to A$, and the **value function** for a policy π, evaluated at any state s_1 is given by the following

$$V^{\pi}(s_1) = \mathrm{E}\left[R(s_1) + \gamma R(s_2) + \gamma^2 R(s_3) + \cdots | \pi\right]$$

Where the expectation is over the distribution of the state sequence $s_1, s_2, \ldots)$which passes through when the policyπ starting from s_1,. then they defined the Q-funciton as follows:

$$Q^{\pi}(s, a) = R(s) + \gamma \mathrm{E}_{s' \sim P_{sa}(\cdot)}\left[V^{\pi}(s')\right]$$

Theorem 1 Bellman equation:

Let an MDP $M = (S, A, \{P_{SA}, r)$ and a policy $\pi : S \to A$ be geive then for all $s \epsilon S, a, \epsilon A, V^{\pi}$ and Q^{π} to satisiy

$$V^{\pi}(s) = R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^{\pi}(s') \quad (1)$$

$$Q^{\pi}(s, a) = R(s) + \gamma \sum_{s'} P_{sa}(s') V^{\pi}(s') \quad (2)$$

Theorem 2 Bellman Optimality

Where ***MDP, $M = (S, A, \{P_{SA}, r)$ and a policy $\pi : S \to A$ be geiven then π is an optimal policy for M if an $s \epsilon S$***

Three following subsections of the paper describe the algorithm and experimentation process implemented on each state space to conduct the experiments and gather data in relation to the optimal policy and finding the true reward as follows.

IRL infinite State spaces

In this study, the researchers characterized the set of all reward functions for which a given policy is optimal. They showed that the set contains many degenerate solutions and proposed a heuristic This theorem uses finite-state MDP which resulted in a character I said of all reinforcement functions that are solutions to the inverse reinforcement learning problem. this test case showed that two problems existed first the reward $R = 0$ being the same no matter what action is taken within any policy and second, it seemed like there were too many choices of R that met the criteria leaving the researchers questioning which reinforcement functions to choose. This experiment was defined on a simple state space which is finite and known. The optimize problem solution is as follows:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{i=1}^{N} \min_{a \in \{a_2, \ldots, a_k\}} \{(\boldsymbol{P}_{a_1}(i) - \boldsymbol{P}_a(i)) \\ & (\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R}\} - \lambda ||\boldsymbol{R}||_1 \\ \text{s.t.} \quad & (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \succeq 0 \\ & \forall a \in A \setminus a_1 \\ & |\boldsymbol{R}_i| \leq R_{\max}, \quad i = 1, \ldots, N \end{aligned}$$

Linear function approximation in large State spaces

The researchers then implemented the case for infinite State spaces which was defined in the same way as finite state space MDP where $S = R$ as a set of real numbers and the availability subrutine for approximation the value policy $V^{\pi}$for any particular MDP. This linear programming formulation was defined as follows:

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

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{s \in S_0} \min_{a \in \{a_2, \ldots, a_k\}} \{ \\ & p(\mathrm{E}_{s' \sim P_{sa_1}}\left[V^{\pi}(s')\right] - \mathrm{E}_{s' \sim P_{sa}}\left[V^{\pi}(s')\right])\} \\ \text{s.t.} \quad & |\alpha_i| \leq 1, \quad i = 1, \ldots, d \end{aligned}$$

IRL from sample trajectories

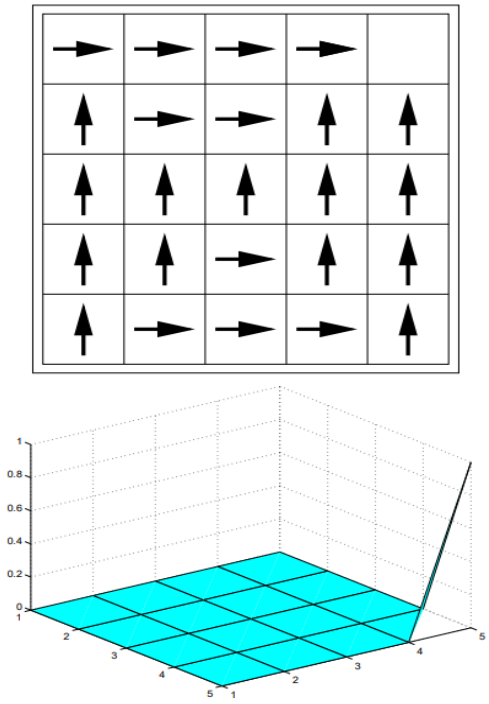
This research addresses the IRL problem for the more realistic case where the researchers had access to the policy only through a set of actual trajectories in the state space; this implementation did not require an explicit model of the MDP. This was assumed to allow them the ability to find an optimal policy under any reward they choose. The algorithm first found the value estimates for the assumed optimal policy which were randomly chosen. An inductive

step to the algorithm defined having some set of policies and finding the α statistical significance of a result so that the resulting reward function would be satisfied.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{k} p\left(\hat{V}^{\pi^*}(s_0) - \hat{V}^{\pi_i}(s_0)\right) \\ \text{s.t.} \quad & |\alpha_i| \leq 1, \quad i = 1, \ldots, d \end{aligned}$$

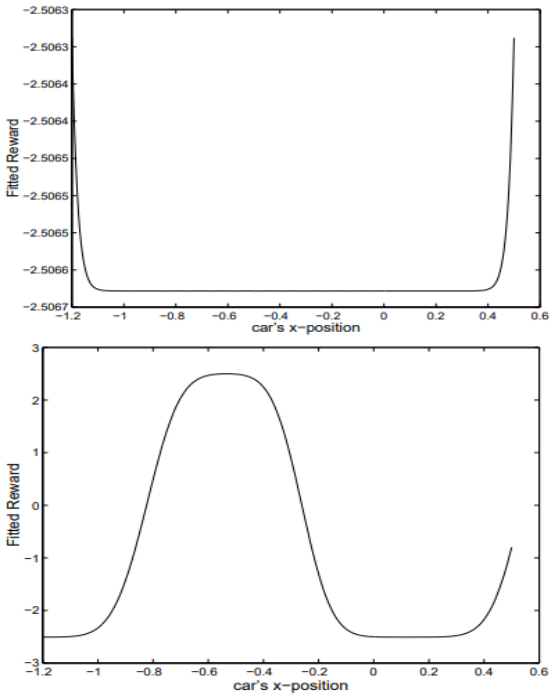
III. RESULTS AND DISCUSSION

As shown in Fig1 below, the first experiment uses a 5 x 5 grid world, where a reward is given for moving in one of the four compass directions but is noisy and has 30% chance of moving in a random direction instead, and the optimal policy is displayed along with the IRL state space together with the true reward function, they found the inverse reinforcement problem is that of recovering the word structure given the policy in the problem dynamics [1].



As shown in figure 2 above, doing the algorithm described in section 3.2 of the research paper with no penalty term yields a reward function that is not concise; it is also found to have issues from the arbitrary symmetry breaking in the chosen policy. However, with the penalty efficiently set to a value just below the phase transition they were able to obtain the second reward function which is very close to a tree reward.

Fig. 3 shows a cartoon of another well-known mountain cart, an experiment where the true undiscounted reward is -1 first Step until they reach the goal at the top of the hill in the state is the car's x-position and base velocity since the state space is continuous. The hill in the state is the car's x-position and base velocity since the state space is continuously distributed. The researchers then present an approximator class for the reward function to be based on the car's x-position only, with the function class consisting of 26 evenly spaced gaussian functions. x-position of the car only, made up of 26 evenly spaced Gaussian



They also re-ran the with the true reward change to be 1 in an interval [-0.72,-0.32] centered around the bottom of the hill and 0 everywhere else with a gamma of 0.9. In this problem the optimal policy is to go as quickly as possible to the bottom of the hill and park there. This was not always possible because the example showed they were near the top of the hill on the right and moving too quickly then they shot off to the right end of the hill and entered the absorbing State no matter how hard they broke running the algorithm on this difficult solution. The researchers did find positive conclusions to these results.

The final experiment applied the sample-based algorithm to the newest version of the five-by-five grid more precisely to the state [0,1,0,1] and the effect of each of the four-compass direction action is added to the intended direction after which uniform noise [-0,1,0,1] was added. Experiments showed that most algorithms end up with fairly good solutions; they compare the fit between the algorithm and the research has found discrepancies typically between 3 and 10% with much distinct near-optimal policy such as variation. According to the researchers, a better measure of the algorithm's performance can be comp In retrospect it was noted that it usually took about 15 iterations of the algorithm and the evaluations which use 50,000 Monte Carlo trials at 50 steps each unable to detect statistically significant differences between the value of true optimal policy which was near 6.65.

IV. CONCLUSION

The researchers found that inverse reinforcement learning may be useful for apprenticeship learning used to acquire skilled behavior and for serving their reward function being optimized by a natural system. In this case it would contribute to research that one day some artificial intelligence researchers may achieve finding the True reward and IRL might be one approach to understand what humans want and to hopefully work towards these goals. While none of the experiments in this research were able to validate sufficient evidence or solve the IRL problem. The results show that the inverse reinforcement learning problem is soluble for moderate size discrete-continuous domains which left the researchers with some very important questions left on the table. Some of these questions in regard to agent behavior are strongly inconsistent with optimality. The study proved to leave researchers with some valuable questions such as if they are locally consistent reward functions for specific reasons in a state-space? In real-world application they found that there is too much noise in the agent sensor inputs, essentially based shaping rewards can't produce reward functions that make it dramatically easier to learn a solution to mark of decision process without affecting optimality it's not knowing if they can design IRL algorithms cover is reward functions, how can I experiments be designed to maximize identify the ability of the reward function, and how well does the algorithmic approach carry the case of partially observable environments.

The field of IRL has undergone many advancements since this paper was published. According to a source published in 2020 titled UAV Autonomous Aerial Combat Maneuver Strategy Generation with Observation Error Based on State-Adversarial Deep Deterministic Policy Gradient and Inverse Reinforcement Learning, unmanned aerial vehicles are using inverse reinforcement learning methods to further enhance their artificial intelligence capabilities. They model the aerial combat WVR as a state-adversarial Markov decision process (SA-MDP), which introduces the small adversarial perturbations on state observations and these perturbations did not alter the environment directly, but where used to mislead the agent into making suboptimal decisions. This paper proposes a novel autonomous aerial combat maneuver strategy generation algorithm with high-performance and high-robustness based on state-adversarial deep deterministic policy gradient algorithm (SA-DDPG), which adds robustness regularizes related to an upper bound on performance loss at the actor-network. At the same time, a reward shaping method based on maximum entropy (Maxent) inverse reinforcement learning algorithm (IRL). This research is proposed to improve the aerial combat strategy generation algorithm [3]. Although the problem is not completely solved there are many uses from the advancements in this exciting field

V. REFERENCES

[1] S. R. Andrew Y .NG, "Algorithms for inverse reinforcement learning," *UC Berckley* , vol. 1, no. 1, p. 8, 2000.

[2] S. Arora, "A Survey of Inverse Reinforcement Learning," THINC Lab, Dept. of Computer Science, University of Georgi, Georgia, 2020.

[3] D. Z. Y. Z. a. Z. Weiren Kong *, "UAV Autonomous Aerial Combat Maneuver Strategy Generation with Observation Error Based on State-Adversarial Deep Deterministic Policy Gradient and Inverse Reinforcement Learning," *MDPI*, vol. 1, no. 1, p. 7, 2020.