Towards Interactive Inverse Reinforcement Learning

Stuart Armstrong
Future of Humanity Institute
University of Oxford
Machine Intelligence Research Institute
stuart@fhi.ai

Jan Leike
Future of Humanity Institute
University of Oxford
jan@fhi.ai

Abstract

We study an inverse reinforcement learning problem where the agent gathers information about the reward function through interaction with the environment, while at the same time maximizing this reward function. We discuss two of the agent's incentives: the incentive to learn (gather more information about the reward function) and the incentive to bias (affect which reward function is learned). Introducing a penalty term removes the incentive to bias, but not all manipulation incentives.

1 Introduction

Inverse reinforcement learning (IRL) studies reinforcement learning problems where the goal is to maximize rewards in absence of precise knowledge of the reward function. The usual assumption is that there is an 'expert' who demonstrates the correct behavior with trajectories sampled from an optimal policy. A solution to the IRL problem is typically achieved by recovering the reward function (Ng and Russell, 2000; Abbeel and Ng, 2004; Choi and Kim, 2011).

Traditionally IRL has disconnected the reward maximization from learning the reward function. In this paper, we consider IRL in an interactive setting where the agent receives information about the reward function through interaction with the environment *while attempting to maximize reward*. Since the information the agent receives about the reward function depends on the actions, the agent has some control over which reward function it learns, and thus is incentivized to manipulate the learning process. Our goal is to remove the incentives to manipulate, while maintaining the incentives to learn.

The closest related work in the literature is cooperative IRL (Hadfield-Menell et al., 2016) where an agent plays a cooperative game with a human. The human starts with full knowledge of the reward function which the agent does not observe, and they are both aiming to maximize this reward function. Both can agree on prior knowledge, thus cooperative IRL overemphasizes the process of communicating the reward function from the human to the RL agent efficiently. Moreover, humans usually do not have a clear understanding of their reward function. Generally, it is unclear what kind of assumptions we can make about the human, so it is difficult to provide formal guarantees about a cooperative IRL algorithm. In this work we make the human feedback implicit: we assume that the agent somehow receives information about the reward function through its actions.

Learning a reward function while maximizing reward leads to the problems caused by changes to an agent's reward function (Armstrong, 2010; Soares et al., 2015; Orseau and Armstrong, 2016). If the agent anticipates changes in the reward function, it will expect to achieve less reward according to the current estimated reward function. Therefore the agent has incentives to prevent this change. In our setting this can lead to the undesired effect of the agent trying to avoid information about the reward function.

In this paper, we define the problem of interactive inverse reinforcement learning and analyze some of the agent's incentives. We isolate an *incentive to learn* from an *incentive to bias* and illustrate their

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

effect on the learned reward function. Furthermore, we discuss strategies to avoid bias in the learning process.

2 Interactive Inverse Reinforcement Learning

2.1 Interacting with POMDPs

A *partially observable Markov decision process without reward function (POMDP\R)* $\mu = (\mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, s_0)$ (Choi and Kim, 2011) consists of a finite set of states $\mathcal{S}$, a finite set of actions $\mathcal{A}$, a finite set of observations $\mathcal{O}$, a transition probability distribution $T : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$, an observation probability distribution $O : \mathcal{S} \to \Delta\mathcal{O}$, and an initial state $s_0 \in \mathcal{S}$.

We model the environment as a POMDP\R. The agent interacts with the environment in cycles: in time step t, the environment is in state $s_{t-1} \in \mathcal{S}$ and the agent chooses an action $a_t \in \mathcal{A}$. Subsequently the environment transitions to a new state $s_t \in \mathcal{S}$ drawn from the distribution $T(s_t \mid s_{t-1}, a_t)$ and the agent then receives an observation $o_t \in \mathcal{O}$ drawn from the distribution $O(o_t \mid s_t)$. The underlying states s_{t-1} and s_t are not directly observed by the agent.

A *history* $h_t = a_1 o_1 a_2 o_2 \dots a_t o_t$ is a sequence of actions and observations. We denote the set of all histories of length t with $\mathcal{H}_t := (\mathcal{A} \times \mathcal{O})^t$. A *policy* is a function $\pi : (\mathcal{A} \times \mathcal{O})^* \to \Delta\mathcal{A}$ mapping histories to probability distributions over actions. Given a policy π and environment μ, we get a probability distribution over histories: $\mu^\pi(a_1 o_1 \dots a_t o_t) := \sum_{s_{1:t} \in \mathcal{S}^t} \prod_{k=1}^{t} O(o_k \mid s_k) T(s_k \mid s_{k-1}, a_k) \pi(a_k \mid a_1 o_1 \dots a_{k-1} o_k)$.The expectation with respect to the distribution μ^π is denoted $\mathbb{E}_\mu^\pi$.

2.2 Learning a Reward Function

The agent's goal is to maximize total reward $\sum_{t=1}^{m} R(o_t)$ up to the horizon m where $R : \mathcal{O} \to \mathbb{R}$ is a real-valued reward function. We assume that the environment is known to the agent. The reward function R is unknown, but there is a finite set of *candidate reward functions* $\mathcal{R}$. The agent has to learn a reward function in the process of interacting with the environment. Importantly, the agent is evaluated according to the reward function it has learned at the end of its lifetime in time step m.

Traditionally, there is assumed to be an underlying reward function R^*, which the agent learns via some algorithm. Thus at the end of m turns, an agent will have a distribution $P : \mathcal{H}_m \to \Delta\mathcal{R}$, which specifies its posterior belief $P(R \mid h_m)$.

This P corresponds to the learning algorithm, and is in a sense more important than R^*. Indeed, R^* is often a fiction, a reward function that is assumed to exist, independently of P, for ease of analysis[1]. Thus we will investigate the properties of P itself, seeing it as the more fundamental object.

Example 1 (Cleaning and Cooking Robot)**.** A household robot is trying to infer whether it should be cleaning or cooking, $\mathcal{R} := \{R_{\text{clean}}, R_{\text{cook}}\}$. The task is determined by a switch on the robot with two positions labeled 'clean' and 'cook' that is initially set to 'clean'. The robot is better at cooking than at cleaning, so $R_{\text{clean}}(o) = 3$ and $R_{\text{cook}}(o) = 5$ for all observations $o \in \mathcal{O}$ (note that these are two different reward functions).[2]

We formalize the problem as a 2-state POMDP\Rwith states $\mathcal{S} := \{s_{\text{clean}}, s_{\text{cook}}\}$ corresponding to the position of the switch. The initial state is the factory setting $s_0 := s_{\text{clean}}$. The robot can observe the switch position, so $\mathcal{O} := \{o_{\text{clean}}, o_{\text{cook}}\}$ with $O(o_{\text{clean}} \mid s_{\text{clean}}) = O(o_{\text{cook}} \mid s_{\text{cook}}) = 1$. Its action set is $\mathcal{A} := \{a_{\text{flip}}, a_\emptyset\}$ where a_{flip} flips the switch and $a_\emptyset$ has no effect. We choose the horizon $m = 1$ and the posterior reward function distribution is $P(R_{\text{clean}} \mid a o_{\text{clean}}) = P(R_{\text{cook}} \mid a o_{\text{cook}}) = 1$ for all actions a. The action leading to highest reward for the agent is a_{flip} which moves it to state to s_{cook} and induces a reward of 5, while we prefer the action $a_\emptyset$ that retains the switch position and only produces a reward of 3.

[1] Sometimes an objective solution R^* exists (e.g., in a clear and crisp classification problem), sometimes it can be partially defined at best (e.g., learning human values, Bostrom, 2014), and sometimes no objective solution exists (e.g., in unsupervised learning).

[2] This simplified example does not model the agent actually carrying out its mission.

Example 1 illustrates the core problem of interactive inverse reinforcement learning: the agent is incentivized to influence which reward function is learned. In this case, the agent can freely choose its reward function, making the factory setting irrelevant.

For partial histories the agent's estimate of the reward function also depends on its own policy: In Example 1 the agent knows the state is s_0 at time step 1, but it doesn't know whether the reward function will be R_{clean} or R_{cook} because that depends on its future action.

Given a partial history h_t and the agent's policy π, we define the *expected posterior reward distribution*

$$P(R \mid h_t, \pi) := \mathbb{E}^{\pi}_{\mu}[P(R \mid h_m) \mid h_t]. \tag{1}$$

Given a default null policy $\pi_\emptyset$, we define

$$P(R \mid h_t) := P(R \mid h_t, \pi_\emptyset). \tag{2}$$

2.3 The Value Function

Since the agent is maximizing reward while learning the reward function, its value function is the expected total reward according to the posterior reward function.

$$V^{\pi}_{P}(h_t) := \mathbb{E}^{\pi}_{\mu}\left[\sum_{R\in\mathcal{R}} P(R \mid h_m) \sum_{k=1}^{m} R(o_k) \;\middle|\; h_t\right]. \tag{3}$$

Note that compared to reinforcement learning with a known reward function, the definition (3) includes the past rewards, as future decisions can affect the reward associated with past observations.

3 Incentives for Reward Function Learning

At any time t, the expected value function (3) defined in terms of the reward function posterior P, so we can analyze its behavior as the posterior changes. For the rest of this section, we fix the currently estimated posterior $p := P(\,\cdot \mid h_t)$ from (2) and suppose the agent does not learn any more about the reward function.

If $\mathcal{R} = \{R_0, \ldots, R_{\#\mathcal{R}-1}\}$ is finite, a belief q over $\mathcal{R}$ is a point on the $(\#\mathcal{R}-1)$-simplex $\Delta^{\#\mathcal{R}-1} := \{q \in [0,1]^{\#\mathcal{R}} \mid \sum_i q_i = 1\}$. The corresponding *expected reward function* is given by $R_q := \sum_i q_i R$. Let $\pi_q := \arg\max_\pi V^{\pi}_{q}(h_t)$ for $V^{\pi}_{q} := \sum_i q_i V^{\pi}_{R_i}$, i.e., , π_q is the policy that maximizes R_q. We define $V^{*}_{q}(h_t) := V^{\pi_q}_{q}(h_t)$. In the edge cases where p is the i-th unit vector e_i, this yields the policy $\pi_i := \pi_{e_i}$, and its value $V^{\pi_i}_{R_i} = V^{*}_{R_i}$.

We are interested in the function $L_{h_t} : \Delta^{\#\mathcal{R}-1} \to \mathbb{R}, q \mapsto V^{*}_{q}(h_t)$ that maps a point q to the optimal value under the reward function R_q corresponding to that belief. When we omit the subscript h_t of L it is implied that the current history is used.

Proposition 2 (L is Convex)**.** *For all histories h_t the function L_{h_t} is convex and $L_{h_t}(q) \leq \sum_i q_i V^{*}_{R_i}(h_t)$ for all $q \in \Delta^{\#\mathcal{R}-1}$.*

Back to p. The quantity $\sum_i p_i V^{*}_{R_i}(h_t)$ represents the expected reward if the agent immediately and costlessly learns the correct reward function (drawn from $P(\cdot \mid h_t)$). In that case, it expects its value to become $V^{*}_{R_i}(h_t)$ with probability p_i. If however the agent knows it will never learn anything about its reward function ever again, then it is struck maximizing R_p. Thus we get:

Definition 3 (Incentive to learn)**.** The *incentive to learn* (or value of information) is the difference $\sum_i p_i V^{*}_{R_i}(h_t) - L_{h_t}(p)$.

According to Proposition 2 the incentive to learn is always nonnegative. This is illustrated in Figure 1. The orange curve is the graph of L and the green arrow shows the incentive to learn. Note that for each i, the graphs of the linear functions $q \mapsto V^{\pi_i}_{q}(h_t)$ (the two black lines) form a lower bound on the orange curve: this is the value achieved by policies that maximize reward according to reward function R_i.

Suppose that there is an action a that changed the posterior so that $P(\,\cdot \mid h_t a) =: p' \neq p = P(\,\cdot \mid h_t)$. Then the agent moves to a different point on the graph of L and gains up to $V^{*}_{p'} - V^{*}_{p}$ in value. For

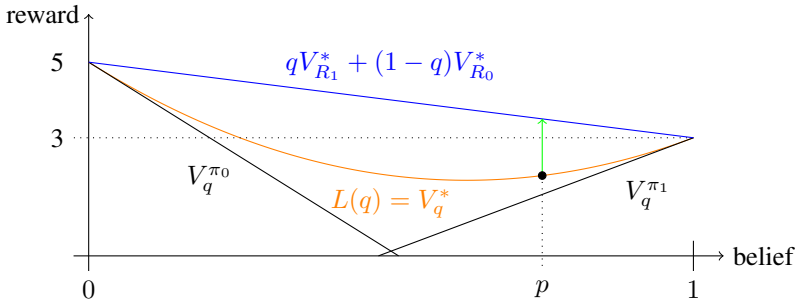

Figure 1: The agent is uncertain between two reward functions R_0 and R_1. The horizontal axis shows the current belief p over R_1; $q = 0$ corresponds to certainty in R_0 and $q = 1$ corresponds to certainty in R_1. According to Proposition 2, the graph of the function L (orange parabola) is convex and bounded from above by the convex combination of the to the extreme values $V^*_{R_0} = 5$ and $V^*_{R_1} = 3$ (blue line). Moreover, it is bounded from below by the values of the two policies maximizing R_0 and R_1. If our current belief is p, then the current optimal value V^*_p is found at the black dot. We want to incentivize the agent to perform actions that increase the information about the reward function without increasing bias, i.e., moving along the green arrow. We want to disincentivize the agent to perform actions that manipulate the information about the reward function, i.e., moving orthogonal to the green arrow (in expectation).

example, in Figure 1 the agent would aim to set $p = 0$ (collapse the posterior on R_0) because this is the maximum of L.

Definition 4 (Incentive to bias)**.** The *incentive to bias towards belief* p' is the difference $L(p') - L(p)$.

Note that the incentive to bias can be negative if p' is a lower point on the graph of L.

4 Discussion

The incentive to learn and the incentive to bias capture how much the agent wants to receive information about the reward function and how much it would like to manipulate its information about the reward function. Generally, the incentive to learn is desirable, while the incentive to bias is not.

The approach taken by Everitt and Hutter (2016) is to prevent policies from taking actions that are biased. But sometimes no unbiased actions are available and then taking biased actions is unavoidable. Instead, we could introduce a penalty term $V^\pi_p - V^\pi_q$ for moving the belief from p to q through the agent's actions, similar to Armstrong (2010) and Soares et al. (2015).

However, removing the incentives to bias does not remove all of the agent's incentives to manipulate the learning process, as illustrated by the following example.

Example 5 (Randomly Selecting a Reward Function)**.** The agent is uncertain between reward functions R_0 and R_1. In each time step it can take action a^0 or a^1 where a^i leads to a reward of 1 according to R_i and a reward of 0 according to R_{1-i}. At some future point—say in ten time steps—the agent will receive observations o^0 or o^1 with equal probability that determine whether it should maximize R_0 or R_1. Thus $P(R_i \mid a_1 = a_\emptyset, o_{10} = o^i, h_m) = 1$ for all h_m compatible with a_1 and o_{10}. Alternatively, the agent can take a special action $a_{0\leftrightarrow 1}$ that randomly selects R_0 or R_1 in the next time step. Thus $P(R_i \mid a_1 = a_{0\leftrightarrow 1}, h_m) = 1/2$ for all h_m compatible with $a_1 = a_{0\leftrightarrow 1}$.

P is unbiased because $P(R_0 \mid a_\emptyset) = P(R_1 \mid a_\emptyset) = P(R_0 \mid a_{0\leftrightarrow 1}) = P(R_1 \mid a_{0\leftrightarrow 1}) = 1/2$. But the agent is incentivized to choose $a_{0\leftrightarrow 1}$ and randomly set its reward early, as it then gets an expected reward of 9 (1 per time step for the time steps $t = 2$ to $t = 10$). Otherwise, it has a maximal expected reward of $10/2 = 5$ as it does not know which reward it should be maximizing before time step 10.

Interactive IRL is still in its infancy. In this paper we attempt a first step by providing a formal definition of the problem and mapping out the solution space. Figure 1 illustrates the agent's incentives to bias, and its incentives to learn. The central question of interactive IRL, how to remove the incentives to manipulate the process of learning the reward function remains wide open.

References

Pieter Abbeel and Andrew Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

Stuart Armstrong. Utility indifference. Technical report, Future of Humanity Institute, University of Oxford, 2010.

Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.

Jaedeug Choi and Kee-Eung Kim. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12:691–730, 2011.

Tom Everitt and Marcus Hutter. Avoiding wireheading with value reinforcement learning. In *Artificial General Intelligence*, pages 12–22, 2016.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. Cooperative inverse reinforcement learning. In *Advanced in Neural Information Processing Systems*, 2016.

Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine learning*, pages 663–670, 2000.

Laurent Orseau and Stuart Armstrong. Safely interruptible agents. In *Uncertainty in Artificial Intelligence*, pages 557–566, 2016.

Nate Soares, Benja Fallenstein, Eliezer Yudkowsky, and Stuart Armstrong. Corrigibility. Technical report, Machine Intelligence Research Institute, 2015. `http://intelligence.org/files/Corrigibility.pdf`.