The Role of Reflexive Reinforcement Learning in Solutions to the Inverse Problem

B. I. Lyons J. Michael Herrmann

***Abstract*—Inverse blah blah**

I. Introduction

Reinforcement learning (RL) [14] has consistently proven itself in a wide variety of simulated tasks, and important intelligence milestones, from competing at a level superior to the greatest human players in traditional board games [5], [12], and competing at an expert level or greater in computer games of varying complexity [4], [9], as well as the learning of complex behaviours in real robot platforms [2], [10], [3], [7] and yet, real world applications of reinforcement learning are few and far between.

Challenges that impact the implementations of real-world RL are varied, though some of the most commonly cited issues are:

- Learning time is impractical, the agent must be able to learn on the real system from a limited number of samples;
- Safety is of paramount importance, and trial and error learning can result in dangers to both the machine and humans in contact with the agent;
- Desires for explainable policies and actions;
- Reward functions that result in representations of the value function which are close to the true value function are hard to define and can result in unforseen abuses of the environment and behaviours which are undesirable.

In recent years, in an attempt to solve or mitigate some or all of these issues, greater interest has been placed in the subfield of RL, *imitation learning*, in which an agent attempts to learn the task either being operated by an expert, or by observing an expert and attempting to replicate the behaviour. One of the most promising avenues for this has been research into inverse reinforcement learning (IRL) [11], in which, an agent attempts, from a collection of trajectories assumed to be near optimal, to reconstruct the reward function implied by the action selection in the state. The motivations for inverse RL are clear both within the field of computer science, and without:

- Modeling of animal and human behaviour for the purposes of scientific enquiry, e.g. if inverse RL is possible we may be able to determine the reward functions utilised by animal behaviours such as foraging and vocalisation in bees and songbirds respectively [11];
- Imitation learning through inverse reinforcement learning such that an agent can learn over a smaller number of samples and at a reduced risk to both itself and humans in the loop;
- Modelling other agents in the environment in which the agent exists, both adversarial and cooperative, which may lead to a better foundation for multi agent tasks.

Whilst IRL is highly promising, it is not without serious drawbacks of its own. It has been noted by many within the field of RL to be an *ill-posed problem* [1], [15], [6], both due to its nature, as well as being computationally intractable, where many possible reward functions can represent the supplied trajectories, and the difficulty of solving partially observable Markov decision processes (POMDP) with a belief state.

It is our research aim to address some of these issues inherent in the IRL problem with our own approach to reinforcement learning, *reflexive reinforcement learning* (RRL) [8], in which an agent is able to adapt its behaviour episode by episode, using a reflexive reward in addition to the task reward, such that it performs the task at a sub-optimal level whilst demonstrating the task to another agent. These tasks have been performed in gridworld environments with the aim to be extended in future work. This instantiation of the RRL paradigm is denoted IRL-RRL, due to using IRL generated information as a reflexive component.

The rest of this paper is organised as follows, in Sect. II we will discuss the inverse reinforcement learning problem, specifically maximum entropy inverse reinforcement learning, as well as the reflexive reinforcement learning paradigm. In Sect. III we discuss the experimental setup, and the metric we use for analysis, before discussing results in Sect. IV. Applications, conclusions, and future work will be discussed in Sections V, VI, VII respectively.

II. Background

A. Preliminaries

A finite Markov decision problem (MDP) is a tuple, (X, A, T, D, R), where X is a finite set of states, A a finite set of actions, $T = P_{x,a}$ such that $P_{x,a}$ is the distribution when taking action a in state x, D is the initial-state distribution from which the initial state x_0 is drawn, and R is a function that assigns to each state-action pair[1] a number r (or a random variable with mean r) which provides a direct or delayed (stochastic) evaluation of this pair.

The RL agent then aims at finding solutions to the Bellman problem of reconstructing a value function that obeys

$$V^*(x) = \max_a \left(R(x,a) + \gamma \sum_{x'} P(x'|x,a) V^*(x') \right) \quad (1)$$

where x is the current and x' the subsequent state, and action a is executable in state x.

This task can be achieved by the assumption that there exists some function $Q^*(x,a)$ which is the value of taking an action in a given state, and that contains the information about the expected reward. In this function approximation approach, we aim to adjust our policy π such that $Q^*(x,a) \leftarrow Q^\pi(x,a), V^*(x) \leftarrow Q^\pi(x)$ by defining them as

$$Q^\pi(x,a) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x, a_0 = a\right] \quad (2)$$

$$V^\pi(x) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x\right] \quad (3)$$

[1] or possibly to triplets: state, action and following state

where $Q^{\pi}(x,a)$ is updated by [13]

$$\Delta Q^{\pi}(x,a) = \alpha \left(r_t + \gamma V(x_{t+1}) - Q(x,a) \right) \tag{4}$$

with discount γ and learning rate α and the value is given by

$$V^{\pi}(x_{t+1}) = \max_{a} Q^{\pi}(x_{t+1}, a). \tag{5}$$

the resulting effect is that the trial and error nature results in $t \to \infty, Q^{\pi}(x,a) \to Q^{*}(x,a), V^{\pi}(x) \to V^{*}(x)$.

B. Inverse Reinforcement Learning

In the case of the IRL problem, we consider an MDP without a reward function, by convention denoted MDP\R, i.e., a tuple of the form (X, A, T, D). We then make some basic assumption: (1) that there exists some function $\phi(x_i) : x_i \to \mathbf{f}_{x_i}$ which linearly maps $x_i \in X$ to some features of the state $\mathbf{f}_{x_i} \in [0,1]^k$, where k is the number of features; (2) there exists some reward function $R^{*}(x) = w^{*} \cdot \phi(x)$, where $w^{*} \in \mathbb{R}^k$ are the reward weights.

The setup then is that there is an (assumed) expert agent which generates trajectories, ζ in each episode, consisting of states x_i and actions a_i, where $i = 1, ..., t$, where t is the length of the episode, and is not necessarily of constant length. Thus, the reward value of a trajectory is the sum of the state rewards

$$R * (\mathbf{f}_{\zeta}) = \sum_{x_i \in \zeta} {w^{*}}^{\top} \mathbf{f}_{x_i} \tag{6}$$

with empirical feature count

$$\mathbb{E}[\mathbf{f}] = \frac{1}{N} \sum_{i}^{N} \mathbf{f}_{\zeta_i} \tag{7}$$

Observing the above it becomes readily apparent that this is an ill-posed problem, as many reward weights will result in the demonstrated trajectories being optimal.

As an alternative, it was proposed in [1] to instead match *feature expectations* estimated by

$$\hat{\mu}(\pi) = \frac{1}{N} \sum_{i}^{N} \sum_{t} \gamma^{t} \mathbf{f}_{t} \tag{8}$$

between an observed policy π and an agents behaviour, where γ is the discount factor, and t is the time step in a given trajectory. Despite this improvement, the problem remains, in that each policy can be optimal for many reward functions, and many policies will lead to the same feature counts.

C. Maximum Entropy Inverse Reinforcement Learning

In an effort to resolve the ambiguities inherent in the IRL problem, Ziebart et al. [15] utilise maximum entropy to reduce to a single stochastic policy over feature counts.

$$P(\zeta | w^{*}, T) = \sum_{o \in \mathcal{T}} P_T(o) \frac{e^{{w*}^{\top} \mathbf{f}_{\zeta}}}{Z(w^{*}, o)} I_{\zeta \in o} \tag{9}$$

$$\approx \frac{e^{{w*}^{\top} \mathbf{f}_{\zeta}}}{Z(w^{*}, T)} \prod_{x_{t+1}, a_{t+1}, x_t \in \zeta} P_T(x_{t+1} | a_t, x_t) \tag{10}$$

with

$$P(a | w^{*}, T) \propto \sum_{\zeta : a \in \zeta_{t=0}} P(\zeta | w^{*}, T) \tag{11}$$

D. Reflexive Reinforcement Learning

RRL considers that the vast quantity of information generated over the learning process has significant value, and that reduction to some pure task policy is a hindrance in attempts at autonomous, continuous learning of an agent. Instead, by utilising a *reflexive component* which adapts the reward signal between the environment and the agent, the state valuation can be informed by other information, with the intention of generating behaviours and valuations which represent the states beyond just a task. Previously we have shown that we are able to generate empowerment [**?**] like quantities on-line [8].

Although in this instance we utilise the Q-learning algorithm as outlined in Sect. II-A, as RRL adjusts the reward signal based on some predetermined criteria, it has the advantage of being compatible with other approaches to RL, as well as being highly flexible.

We believe RRL to be a good approach to improving the efficacy of IRL, as an agent that is “educating" another agent by supplying trajectories should not simply provide the optimal policy, but must instead attempt to accurately depict the reward function of the environment. As such this means continuously adapting its behaviour over a set of trajectories such that the output of the Max Ent IRL algorithm best represents the true reward function, whilst also continuing to perform the task to completion.

III. Methods

A. Environments

As this is a proof of concept we have begun with the two most simple environments.

1) Linear World: The linear world environment is a discrete world of length N, each consisting of two rewards arranged at each of the terminal states, situated at either end of the linear world. In each instance, one reward is greater to or less than the other terminal state to some degree.

2) 2D World: The 2D world environment is a discrete world NxN world, with two to four terminal, reward states at each corner of the state space. In each instance the rewards are determined to be $r_1 \leq r_2 \leq r_3 \leq r_4$.

B. Reflexive Component

1) Two Rewards:

$$R(x,a) = \begin{cases} r_t & \text{if } x \in \mathbb{Q} \\ r_t + \|L\| & \text{if } x \notin \mathbb{Q} \end{cases} \tag{12}$$

where r_t is the task reward and

$$L = \frac{w_{e_s}}{w_{e_l}} - \frac{w_{T_s}}{w_{T_l}} \tag{13}$$

with w_{e_s}, w_{e_l}, are the maximum entropy reward weights predicted for the episode, e, both small, s, and large, l, and w_{T_s}, w_{T_l} are the true weights, T.

2) More Rewards:

Algorithm 1 Reflexive Reinforcement

```
Require: Reflexive policy $\pi_\theta(a|x) = p(a|x, \theta)$, with initial parameters $\theta = \theta_0$ generated by expert learner, reflexive component $f(r) : r \to \hat{r}$
while $e < E$ do
    Draw starting state $x_0 \sim p(x)$
    while $t < T$ do
        Draw action $x \sim p(a|x, \theta)$
        Observe next state $x_{t+1} \sim p(x_{t+1}|x_t, a_t)$
        Observe task related reward $r$
        Reflexive reward $\hat{r} = f(r)$
        $Q(x, a) \leftarrow \alpha\,(r_t + \gamma V(x_{t+1}) - Q(x, a))$
    end while
    $e \leftarrow e + 1$                ▷ continue to next episode
end while
```

C. Success Metrics

D. IRL-RRL

IV. Results

V. Applications

A. Rapid Prototyping

VI. Conclusions

VII. Future Work

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[3] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine learning*, 23(2):279–303, 1996.

[4] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[5] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

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

[7] Napat Karnchanachari, Miguel Iglesia Valls, David Hoeller, and Marco Hutter. Practical reinforcement learning for mpc: Learning from sparse objectives in under an hour on a real robot. In *Learning for Dynamics and Control*, pages 211–224. PMLR, 2020.

[8] B. Lyons. and J. Herrmann. Reflexive reinforcement learning: Methods for self-referential autonomous learning. In *Proceedings of the 12th International Joint Conference on Computational Intelligence - Volume 1: NCTA,*, pages 381–388. INSTICC, SciTePress, 2020.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[10] Jun Morimoto and Kenji Doya. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, 36(1):37–51, 2001.

[11] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[12] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[13] Richard S Sutton and Andrew G Barto. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

[14] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[15] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.