TOWARD COMPUTATIONALLY EFFICIENT INVERSE REINFORCEMENT LEARNING VIA REWARD SHAPING

Lauren H. Cooke*, **Harvey Klyne***, **Edwin Zhang***
Harvard University

Cassidy Laidlaw
University of California, Berkeley

Milind Tambe, Finale Doshi-Velez
Harvard University

ABSTRACT

Inverse reinforcement learning (IRL) is computationally challenging, with common approaches requiring the solution of multiple reinforcement learning (RL) sub-problems. This work motivates the use of potential-based reward shaping to reduce the computational burden of each RL sub-problem. This work serves as a proof-of-concept and we hope will inspire future developments towards computationally efficient IRL.

1 INTRODUCTION

Inverse reinforcement learning (IRL) is the task of deriving a reward function that recovers expert behavior within an environment (Ng & Russell, 2000) and can be computationally expensive to solve. IRL algorithms typically consist of a loop in which every step requires finding the optimal policy for the current reward estimate (e.g. Abbeel & Ng (2004); Ramachandran & Amir (2007); Wulfmeier et al. (2016)). This means that within a single IRL optimization multiple reinforcement learning (RL) problems need to be solved, each of which may be challenging. One can solve RL tasks by planning actions sufficiently far into the future (Sutton & Barto, 2018), and the necessary planning depth is a measure of the computational challenge of the problem. In the special case where the RL optimization makes use of a sample-based solver, planning depth can be thought of in terms of sample complexity (Kakade, 2003). Previous works have attempted to reduce the overall cost of IRL by deliberately truncating the planning depth, accepting an approximation to the optimal policy at each iteration (MacGlashan & Littman, 2015; Xu et al., 2022).

Since multiple reward functions can encode an optimal policy (Russell, 1998; Cao et al., 2021), we have some choice about what reward function we optimize at each iteration. We envision using potential-based reward shaping (Ng et al., 1999) to reduce the computational cost of each RL sub-problem without altering any of the optimal policies. This itself is too large a goal for the present work, so we focus our efforts on demonstrating a proof-of-concept in a simplified setting. In particular, we examine how sample trajectories from both optimal and random policies may be used to select a potential function for an initial feasible reward (one which encodes the optimal policy), which we call planning-aware reward shaping. Previous work on reward shaping includes Hu et al. (2020); Dong et al. (2020); Cheng et al. (2021); Gupta et al. (2022); De Lellis et al. (2023). To be clear, our present procedure does not directly address the problem of making IRL more computationally efficient, but we hope that the conclusions drawn may inspire future work.

2 PLANNING-AWARE REWARD SHAPING

Suppose we have been given a Markov Decision Process without a reward $\mathcal{M} \setminus \mathcal{R} = (\mathcal{S}, \mathcal{A}, \gamma, P)$, and using optimal trajectories have learned a feasible reward R_0 using some IRL algorithm. We also assume access to a set of trajectories which have selected actions uniformly at random, and we use this additional exploration information to make a one-step adjustment to R_0. This adjustment takes

*Equal contribution. First author order is alphabetical.

the form of a potential function $\Phi : \mathcal{S} \to \mathbb{R}$, with our final reward function estimate taking the form

$$R_\Phi(s,a) := R_0(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)}[\Phi(s')] - \Phi(s). \tag{1}$$

The optimal policies for R_0 and R_Φ are the same for any Φ, and any equivalent reward may be written in the form R_Φ for some Φ (Ng et al., 1999). Crucially, the planning depths associated with rewards R_Φ may differ across choices of potential function Φ. We stress that the goal of this work is to inspire future investigation into computationally efficient IRL, a problem we do not claim to have solved here. Our goal is to choose a shaped reward R_Φ which minimizes a certain bound on an algorithm-agnostic measure of planning depth (Laidlaw et al., 2023). This bound has been found to be strongly correlated with the sample complexities of modern deep learning RL procedures — including DQN (Mnih et al., 2015) and PPO (Schulman et al., 2017) — across a range of tasks. Denote by $Q_\Phi^{\text{rand}}(s,a)$ the Q-function associated with the uniform-at-random policy $\pi^{\text{rand}}(a \mid s) := 1/|A|$, and write $V_\Phi^*(s)$ for the value function associated with the optimal policy π^*. Recall that the optimal policy π^* does not depend on Φ. Following the derivation in Appendix A, we find that our objective is minimized (potentially non-uniquely) by

$$\Phi(s) = \left\{ \max_{a \in A} Q_0^{\text{rand}}(s,a) + V_0^*(s) \right\} / 2. \tag{2}$$

Both V_0^* and Q_0^{rand} are learnable from the expert and random exploration trajectories respectively. However, assuming access to the optimal value function V_0^* trivializes the forward RL challenge (e.g. set $R(s) = V_0^*(s)$ and act greedily over next actions). We anticipate that IRL algorithms may be able to iteratively update the shaping potential Φ based on the current estimates of Q_0^{rand} and V_0^*, which may improve the overall computational efficiency. We further remark that estimating Q_0^{rand} is much easier than estimating V_0^*, so shaping based on (2) might have better finite-sample performance than potentials based on estimates of V_0^* alone.

We demonstrate that an oracle version of our procedure reduces the sample complexity for DQN (Figure 1), which we use as a proxy for planning depth. We evaluate our method in a 5×5 deterministic gridworld since we can easily find the optimal policy, yet DQN struggles and takes tens of thousands of steps to converge (Laidlaw et al. (2023, Tab. G.4)).

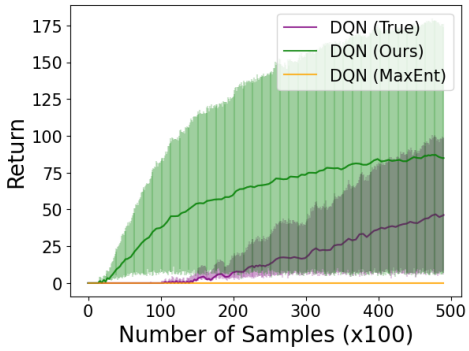

Figure 1: Return (measured by R_0) obtained by DQN. We compare learning with R_0 (purple), R_Φ (green), and R_{MaxEnt} (orange), averaged across 500 random DQN seeds, each training for 50K steps. Our shaped reward R_Φ enables DQN to converge to the optimal value faster than the initial reward R_0, demonstrating a reduction of planning depth. DQN fails to optimize R_{MaxEnt}.

In these experiments, we first fix transition dynamics and an initial reward R_0 before solving for the optimal policy π^* by value iteration. As an IRL baseline, we also perform Maximum Entropy IRL (Ziebart et al., 2008) on π^* to obtain a reward R_{MaxEnt}. We compute the optimal value function V_0^* and the random policy Q-function Q_0^{rand} using Monte Carlo, the potential function Φ using (2), and the shaped reward R_Φ using (1). Note that R_0, R_Φ, and R_{MaxEnt} all encode the same optimal policy. We compare the planning depths of these three rewards using the sample efficiency of DQN, finding that the shaped reward R_Φ enables DQN to converge to the optimal solution faster than the initial reward R_0. We also find that DQN fails to optimize R_{MaxEnt}, with the policy getting stuck in a local optimum state rather than reaching the goal state. Implementation details and code to reproduce our experiments are included in Appendix B and the supplementary materials.

3 Conclusion

In this work, we motivate planning-aware reward shaping to reduce the computational complexity of IRL. Compared to existing IRL approaches, we leverage the additional information included in random trajectories to apply automatic reward shaping. Our hope is that our procedure may inspire novel IRL algorithms which are more computationally efficient. While we focus on the IRL setting, adaptive shaping procedures such as ours may also be of interest to the broader RL community.

URM STATEMENT

The author acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 1–8, 2004.

Haoyang Cao, Samuel N. Cohen, and Łukasz Szpruch. Identifiability in inverse reinforcement learning. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 1–11, 2021.

Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. Heuristic-guided reinforcement learning. *arXiv*, pp. 2106.02757, 2021.

Francesco De Lellis, Marco Coraggio, Giovanni Russo, Mirco Musolesi, and Mario di Bernardo. Guaranteeing control requirements via reward shaping in reinforcement learning. *arXiv*, pp. 2311.10026, 2023.

Yunlong Dong, Xiuchuan Tang, and Ye Yuan. Principled reward shaping for reinforcement learning via Lyapunov stability theory. *Neurocomputing*, 393:83–90, 2020.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham M. Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *arXiv*, pp. 2210.09579, 2022.

Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping. *arXiv*, pp. 2011.02669, 2020.

Sham M. Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, pp. 1412.6980, 2014.

Cassidy Laidlaw, Stuart Russell, and Anca Dragan. Bridging RL Theory and Practice with the Effective Horizon. *arXiv*, pp. 2304.09853, 2023.

James MacGlashan and Michael L. Littman. Between imitation and intention learning. In Qiang Yang and Michael Wooldridge (eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3692–3698, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–670, 2000.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcment learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2586–2591, 2007.

Stuart Russell. Learning Agents for Uncertain Environments (Extended Abstract). In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 101–103. Association for Computing Machinery, 1998. ISBN 1581130570.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, pp. 1707.06347, 2017.

R.S. Sutton and A.G. Barto. *Reinforcement Learning, second edition: An Introduction*. Adaptive Computation and Machine Learning series. MIT Press, 2018. ISBN 9780262352703.

Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv*, pp. 1507.04888, 2016.

Yiqing Xu, Wei Gao, and David Hsu. Receding horizon inverse reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Proceedings of the Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In Anthony Cohn (ed.), *Proceedings of the 23rd National Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

APPENDIX

A REWARD SHAPING OPTIMIZATION OBJECTIVE

Laidlaw et al. (2023) consider algorithm-agnostic proxies for the sample complexities of modern deep RL approaches across measures of correlation, tightness, and accuracy. They introduce the effective horizon $H := \min_k k + \log_{|A|} m_k$, where k is a tuning parameter in a simple Monte Carlo algorithm (see Laidlaw et al. (2023, Alg. 1)) and m_k is the minimum sample size this algorithm requires to find an optimal policy with probability at least $1/2$. It is not feasible to compute H in closed form, but they find that a particular bound serves as a good proxy (Laidlaw et al. (2023, Thm. 5.4)). Considering an MDP with finite-time horizon T and $\gamma = 1$, it holds that:

$$H \leq \min_{k=1,\dots,T} k + \max_{t\in T, s\in S, a\in A} \log_{|A|}\left(\frac{Q_t^k(s,a)V_t^*(s)}{\Delta_t^k(s)^2}\right) + \log_A 6\log(2T|A|^k),$$

where

$$\Delta_t^k(s) = \max_{a\in A} Q_t^k(s,a) - \max_{a'\notin \arg\max_a Q_t^k(s,a)} Q_t^k(s,a'),$$

and Q_t^k, V_t^*, and Δ_t^k are defined as in Laidlaw et al. (2023).

We relax this bound by fixing $k = 1$ — which we think is reasonable considering how k is fixed to 1 in practice (e.g. Laidlaw et al. (2023, Sec. F.1)) — but the potential function Φ we derive generalizes to other choices of k. We consider MDPs with $T = \infty, \gamma < 1$ and time-invariant policies, motivating the following optimization for our planning-aware reward shaping:

$$\max_{\Phi:\mathcal{S}\to\mathbb{R}} \left\{ \max_{s\in\mathcal{S}, a\in A} \log_{|A|}\left(\frac{Q_\Phi^{\text{rand}}(s,a)V_\Phi^*(s)}{\Delta_\Phi(s)^2}\right)\right\}. \tag{3}$$

This is strictly increasing in the following criteria:

$$\ell(\Phi; R_0) := \max_{s\in\mathcal{S}, a\in A} \frac{Q_\Phi^{\text{rand}}(s,a)V_\Phi^*(s)}{\Delta_\Phi(s)^2}.$$

In fact, for any policy π the associated Q-function and value function transform linearly under reward shaping (Ng et al., 1999, Cor. 2):

$$Q_\Phi^\pi(s,a) = Q_0^\pi(s,a) - \Phi(s); \quad V_\Phi^\pi(s) = V_0^\pi(s) - \Phi(s).$$

Therefore $\Delta_\Phi(s) = \Delta_0(s)$ for all potentials Φ, so the objective reduces to

$$\ell(\Phi; R_0) = \max_{s\in\mathcal{S}, a\in A} \frac{\left\{Q_0^{\text{rand}}(s,a) - \Phi(s)\right\}\left\{V_0^*(s) - \Phi(s)\right\}}{\Delta_0(s)^2}.$$

The potential function (2) solves this quadratic for every $s \in \mathcal{S}$, and is thus a global minimizer. The solution can be found by straightforwardly taking the derivative of Equation 3 and setting to 0.

B IMPLEMENTATION DETAILS FOR EXPERIMENTS

In Figure 1 we plot the returns achieved by each optimization procedure at each training episode by taking an average across 500 random seeds, along with 95% bootstrapped confidence intervals. Each seed determined one individual training process, wherein we train for 500 episodes of 100 steps each, for a total of 50K training steps. Returns are evaluated at the end of each episode with respect to the initial reward function R_0, regardless of which reward function is used during training. This ensures that returns are comparable between objectives. We set a 100 timestep limit on the environment.

	Hyperparameter	Value
DQN HP	Optimizer	Adam (Kingma & Ba, 2014)
	Critic architecture	MLP
	Critic learning rate	1e-3
	Critic hidden layers	1
	Critic hidden dim	24
	Critic activation function	ReLU
	Mini-batch size	1024
	Number of gradient steps	50K
	Discount factor	0.99
	Target update rate	1
	Target update period	8
	Loss Function	Huber Bellman Error

Table 1: Hyperparameters for the DQN algorithm used in Section 2.

.

To perform DQN we use an MIT-licensed implementation (`github.com/mswang12/minDQN`) with hyperparameters as in Table 1. Code to reproduce our experiments is included in the supplementary materials.