A Novel Inverse Reinforcement Learning Formulation for Sample-Aware Forward Learning

Giorgio Manganini
Department of Computer Science
Gran Sasso Science Institute
L'Aquila, Italy
giorgio.manganini@gssi.it

Angelo Damiani
Department of Computer Science
Gran Sasso Science Institute
L'Aquila, Italy
angelo.damiani@gssi.it

Alberto Maria Metelli
Artificial Intelligence and Robotic Laboratory
Politecnico di Milano
Milano, Italy
albertomaria.metelli@polimi.it

Marcello Restelli
Artificial Intelligence and Robotic Laboratory
Politecnico di Milano
Milano, Italy
marcello.restelli@polimi.it

Abstract

We propose a novel formulation for the Inverse Reinforcement Learning (IRL) problem, which jointly accounts for the compatibility with the expert behavior of the identified reward and its effectiveness for the subsequent forward learning phase. Albeit quite natural, especially when the final goal is apprenticeship learning (learning policies from an expert), this aspect has been completely overlooked by IRL approaches so far. We propose a new model-free IRL method that is remarkably able to autonomously find a trade-off between the error induced on the learned policy when potentially choosing a sub-optimal reward, and the estimation error caused by using finite samples in the forward learning phase, which can be controlled by explicitly optimizing also the discount factor of the related learning problem. The approach is based on a min-max formulation for the robust selection of the reward parameters and the discount factor so that the distance between the expert's policy and the learned policy is minimized in the successive forward learning task when a finite and possibly small number of samples is available. Differently from the majority of other IRL techniques, our approach does not involve any planning or forward Reinforcement Learning problems to be solved. After presenting the formulation, we provide a numerical scheme for the optimization, and we show its effectiveness on an illustrative numerical case.

Keywords: Inverse Reinforcement Learning, Off-line Reinforcement Learning, Sample Complexity, Min-Max Optimization

Acknowledgements

This work was partially supported by the PON - Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, AIM1880573 Smart, Secure and Inclusive Communities.

1 Introduction

Inverse Reinforcement Learning [IRL, 11] is the process of recovering, from (demonstrations of) an expert's policy, a reward function, which in many cases is the most parsimonious way to describe the behaviour of the expert, especially in complex problems. The learned reward is intended to be successively used in forward Reinforcement Learning (RL) to find new policies that could generalize over unseen states or even improve the expert's actions in new environments, accounting for transferability of the expert's intention, which is compactly described by its reward function.

In their quest for the expert's reward function, many IRL approaches implement an iterative process [1, 15, 4, 5], that alternates between solving a forward RL problem and updating a reward function estimate. In particular, consistency in terms of performance equivalence between the demonstrated trajectories and the ones induces by the learner's policy are enforced. Then, the learning of a generalized policy is performed using a forward RL procedure based on the current estimate of the reward. Another main common aspect in most IRL approaches is the assumption about the underlying MDP. Traditionally, the vast majority of IRL methods rely on the knowledge of the model (either given or accurately learnt from the demonstrated trajectories) [11, 15, 10], which sometimes is also used to perform the internal forward RL subroutines for finding/evaluating intermediate optimal policies. More recently model-free approaches have been proposed [5, 9], even though some of them still require continuous interactions with the environment [1, 4, 5].

In this work, we take a different point of view on IRL and focus on finding a reward function not only compatible with the expert's demonstrations, but that can make the next forward learning phase as efficient as possible, in terms of the sample complexity required to learn a near-optimal policy. We explicitly take into account *how* the recovered reward function will be employed, i.e., plugged into (a possibly different) environment and used to perform forward RL. In this spirit, among the compatible ones, we prefer the rewards that make the next forward learning phase as *efficient* as possible. This goal is *indirectly* pursued by many IRL algorithms, but, to the best of our knowledge, no algorithm performs the reward selection phase by *explicitly* quantifying the sample complexity of forward RL. The novel formulation we propose blends these ambitious goals together and results in an algorithmic procedure which *i)* is purely model–free, *ii)* does not need any interaction with the environment to collect new on-policy data for policy evaluation, and *iii)* does not require solving any forward problem (*i.e.*, finding an optimal policy given a candidate reward function).

2 Background

We define a Markov Decision Process as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are continuous state and action spaces, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ is the transition model $P(s'|s, a)$, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ specifies the action density for each state $\pi(a|s)$. We denote with $Q^{\pi}_{r,\gamma} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the state-action value function of the policy π. Any policy $\pi \in \Pi$ satisfying $\int_{\mathcal{A}} \pi(a|s) Q(s, a) \mathrm{d}a = \max_{a \in \mathcal{A}} Q(s, a)$ for all states $s \in \mathcal{S}$ is named greedy w.r.t. the function Q. Given Q, we denote with $\mathcal{G}[Q] \subseteq \Pi$ the set of greedy policies w.r.t. Q.

The IRL problem aims at finding a reward function r that can explain the behaviour of an expert that follows a policy π_E, which is optimal w.r.t. some unknown reward r_E [11]. Formally, a reward r is *compatible* with the expert's policy π_E if $\pi_E \in \Pi$ is optimal under r, *i.e.*, $\pi_E \in \mathcal{G}[Q^{\star}_{r,\gamma}]$, where $Q^{\star}_{r,\gamma}$ is the optimal state-action value function under the pair (r, γ).

3 The IRL formulation for Efficient Forward Learning

The issue of efficient learning is related with the sample complexity of finding a good approximation of the optimal policy. In RL, the number of calls to the sampling model are generally a function of the problem parameters and, in particular, of the discount factor γ (linked to the effective number of decision epochs). The smaller is the discount factor, the smaller is the number of samples required to attain a near-optimal estimate of the optimal value-function, as shown in many sample complexity bounds depending on a power of $1/(1-\gamma)$ [*e.g.* 2]. Moreover, the recovered reward in IRL has to be compatible with expert's demonstration, which are known only within some accuracy ϵ and confidence δ, being estimated from a finite set of demonstrations.

The following novel IRL formulation blends all these elements together, and takes into direct consideration the effect of the learned IRL reward on the subsequent forward learning phase. Suppose we are given a forward RL algorithm $\mathfrak{A}$ that, provided with a reward function r, a discount factor γ, and a number of samples $M \geq 0$ is able to output an $\epsilon_{\mathfrak{A}}(M, \gamma)$-approximation of the expert Q-function, with probability at least $1 - \delta$. Then, the influence of the IRL reward and discount factor on the distance between the expert's policy and the learned policy in the successive forward learning task, when M samples are available, can be captured by the next adversarial min-max optimization program:

$$\min_{r,\gamma} \max_{\pi \in \mathcal{G}\left[\widehat{Q}^{\pi_E}_M\right]} \left\| Q^{\pi_E}_{r_E,\gamma_E} - Q^{\pi}_{r_E,\gamma_E} \right\| \tag{1a}$$

$$\text{s.t.} \left\| \widehat{Q}^{\pi_E}_M - Q^{\pi_E}_{r,\gamma} \right\| \leq \epsilon_{\mathfrak{A}}(M, \gamma), \quad r \in \mathcal{R},\ \gamma \in [0, 1), \tag{1b}$$

where $\mathcal{R}$ is a set of available reward functions and $\|\cdot\|$ is a suitably defined norm.

The formulation (1a) constitutes a worst-case guarantee on the sub-optimality of the learned policy π w.r.t. the expert's policy π_E, when evaluated under the true (and unknown) reward r_E and discount factor γ_E. This implies also the compatibility of the learned reward r with the expert's policy, which is the main requirement in IRL. Moreover, the explicit optimization of the learned discount factor γ allows to trade-off with the reward itself the optimality of the learned policy π, and hence tune the sample complexity in the subsequent forward RL task. To this end, we define in (1b) the confidence region of the estimated expert's Q-function $\widehat{Q}_M^{\pi_E}$ under the optimized reward r and discount factor γ. This set determines the feasible domain where we can seek for a greedy policy mimicking the expert's one, which will be known within some accuracy ϵ and confidence level δ varying with the number of data M available during the forward learning phase.

4 Construction of a Solvable IRL Formulation

This section is devoted to the construction of a numerically tractable optimization problem for the formulation (1), which is not readily solvable because of the unknown quantities r_E and γ_E in its objective function (1a). First, we consider linearly independent features $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to [0,1]^{d_\theta}$ and $\boldsymbol{\psi} : \mathcal{S} \times \mathcal{A} \to [0,1]^{d_\omega}$ through which we linearly parametrize the reward function and the action-value function: $r_{\boldsymbol{\theta}}(s,a) = \boldsymbol{\phi}(s,a)^\top \boldsymbol{\theta}$ and $\widehat{Q}^{\pi}_{\boldsymbol{\omega}}(s,a;\boldsymbol{\omega}) = \boldsymbol{\psi}(s,a)^\top \boldsymbol{\omega}$, where $\boldsymbol{\theta} \in \mathbb{R}^{d_\theta}$ and $\boldsymbol{\omega} \in \mathbb{R}^{d_\omega}$. We restrict the search of the greedy policy over a class of parametric differentiable policies $\Pi_{\boldsymbol{\eta}} = \{\pi_{\boldsymbol{\eta}} : \boldsymbol{\eta} \in \mathbb{R}^{d_\eta}\}$. We manipulate the objective function and the constraint of the formulation (1) according to the following steps.

1) Wasserstein Distance on Expert's Policy π_E

We bypass the value-function representations in (1a), and hence remove the dependence on expert's reward r_E and discount factor γ_E, [1] by considering a surrogate objective function based on the notion of policy divergence. Specifically, we bound the Q-function distance with the policy distance, thanks to the following theorem.

Theorem 4.1. *[12] If the MDP and the policy π_E are Lipshitz continuous with constants (L_r, L_P) and L_π. Then, it holds that*

$$\left\|Q^{\pi_E}_{r_E,\gamma_E} - Q^{\pi}_{r_E,\gamma_E}\right\|_\mu \leq \frac{\gamma_E L_r L_\pi}{(1-\gamma_E)\left(1-\gamma_E L_P(1+L_\pi)\right)} \times \int_{\mathcal{S}\times\mathcal{A}} d^{\pi_E}_{\mu,\gamma_E}(s) W_2(\pi_E(\cdot|s), \pi(\cdot|s))\,\mathrm{d}s\,,$$

where W_2 is the L_2-Wasserstein distance and $d^{\pi_E}_{\mu,\gamma_E}$ is the γ_E-discounted state occupancy induced by policy π_E.

Since we deal with continuous actions and deterministic policies (the expert's policy is usually deterministic), the Wasserstein's distance is an appropriate distributional divergence. Given two deterministic policies $\pi_{\boldsymbol{\eta}}$ and π^E, and a state $s \in \mathcal{S}$, it can be computed as $W_2^2\big(\pi^E(s), \pi_{\boldsymbol{\eta}}(s)\big) = \big(\pi^E(s) - \pi_{\boldsymbol{\eta}}(s)\big)^2$.

2) Dealing with the Forward Q-function $\widehat{Q}_M^{\pi_E}$

We replace, in the constraint (1b), the forward Q-function $\widehat{Q}_M^{\pi_E}$, which is not available during the IRL problem, with one computable during the IRL task, say $\widehat{Q}_N^{\pi_E}$. To this end we start simplifying the greedy constraint from (1a):

$$\pi_{\boldsymbol{\eta}} \in \mathcal{G}\left[\widehat{Q}_M^{\pi_E}\right] \Rightarrow \widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) \geq \widehat{Q}_M^{\pi_E}(s,\pi_E(s)) \quad \forall s \in \mathcal{S} \Rightarrow \sum_{s\in\mathcal{D}_{\text{IRL}}} \widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_M^{\pi_E}(s,\pi_E(s)) \geq 0, \tag{2}$$

The first relaxation involves the transition from a greedy policy to all policy with at least some performance improvement, so as to have an explicit dependence of the learner policy $\pi_{\boldsymbol{\eta}}$ on $\widehat{Q}_M^{\pi_E}$. The second relaxation implies that the constraint should hold on average over a finite subset of selected states $\mathcal{D}_{\text{IRL}} \subseteq \mathcal{S}$, since it would be impossible enforce it in an infinite state space (a similar relaxation is operated for instance in [14] for the KL-divergence).

We now use, in the next proposition (see Appendix A for the proof), the original confidence interval (1b) in combination with the policy improvement inequality (2) to compute a looser constraint than (2) but that does not involve the unknown quantity $\widehat{Q}_M^{\pi_E}$.

Proposition 4.2. *Let $\widehat{Q}_M^{\pi_E}$ be a Q-function known with accuracy ϵ_M (with probability $1-\delta_M$) and let $\widehat{Q}_N^{\pi_E}$ be the Q-function estimated with N samples during IRL with accuracy ϵ_N (with probability $1-\delta_N$), i.e., :*

$$\left\|\widehat{Q}_M^{\pi_E}(s,a) - Q^{\pi_E}_{r,\gamma}(s,a)\right\| \leq \epsilon_M, \quad \left\|\widehat{Q}_N^{\pi_E}(s,a) - Q^{\pi_E}_{r,\gamma}(s,a)\right\| \leq \epsilon_N. \tag{3}$$

Then, with probability at least $1-\delta_M-\delta_N$, inequality (2) is relaxed as:

$$\sum_{s\in\mathcal{D}_{\text{IRL}}} \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) + 2\epsilon_M + 2\epsilon_N \geq 0. \tag{4}$$

[1] The Q-functions refer to the optimized pair (r,γ) which, for compactness, are removed from the subscripts.

As for the parameters ϵ_M and ϵ_N, we now consider the general structures $\epsilon_M = {}^{\gamma c_M}/_{(1-\gamma)\sqrt{M}}$, $\epsilon_N = {}^{\gamma c_N}/_{(1-\gamma)\sqrt{N}}$, which generalize most of the sample complexity bounds available in the literature [*e.g.* 2]. Parameters c_M and c_N are problem and algorithm-specific constants, that we will treat as hyperparameters.

Remark 4.3. We remark that the assumption in (3) is simply the constraint (1b) itself, which we directly use here to build the new relation involving only $\widehat{Q}_N^{\pi_E}$. In particular, inequality (4) comprises all the uncertainties related either with $\widehat{Q}_M^{\pi_E}$ and $\widehat{Q}_N^{\pi_E}$, and it depends on both the outer optimization variables γ (via parameters ϵ_M and ϵ_N, and the Q-function $\widehat{Q}_N^{\pi_E}$) and r (via the Q-function $\widehat{Q}_N^{\pi_E}$), as well as on the number of samples N available for the IRL task (rather then on the number of samples M that will be used in the forward RL problem). Dependence of (4) on the policy $\pi_{\boldsymbol{\eta}}$ is instead straightforward.

3) Expert's Policy Evaluation with $\widehat{Q}_N^{\pi_E}$

The final stage in our construction of a solvable IRL formulation is the estimation of the new Q-function $\widehat{Q}_N^{\pi_E}$. While, in principle, any policy evaluation algorithm may be used to this purpose, here we resort to the the least-squares temporal difference [LSTDQ, 7] algorithm, for which a confidence region of the form (3) is available via Finite-sample Analysis. In particular, the prediction error of the LSTDQ estimation $\widehat{Q}_N^{\pi_E}(s,a;\widehat{\boldsymbol{\omega}})$ w.r.t. the true value function $Q_{r,\gamma}^{\pi_E}(s,a)$ is provided by [8, Theorem 5], with the accuracy parameter ϵ_N asymptotic to ${}^{\gamma c_N}/_{(1-\gamma)\sqrt{N}}$.

5 Optimization Algorithm

Having introduced in the previous section all the necessary elements for the definition of our new IRL formulation, we discuss now the optimzation algorithm for the solution of the min-max optimization problem. We start by rewriting the final optimization problem in terms of the optimization variables $(\boldsymbol{\theta}, \gamma, \boldsymbol{\eta})$ as:

$$\min_{\substack{\boldsymbol{\theta}\in\mathbb{R}^{d_\theta}\\ \gamma\in[0,1)}} \max_{\boldsymbol{\eta}\in\mathbb{R}^{d_\eta}} \underbrace{\sum_{s\in\mathcal{D}_{\text{IRL}}} W_2\big(\pi^E(s),\pi_{\boldsymbol{\eta}}(s)\big)}_{\triangleq f(\boldsymbol{\eta})}, \quad \text{s.t.} \quad \underbrace{\sum_{s\in\mathcal{D}_{\text{IRL}}} \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) + 2\epsilon_M + 2\epsilon_N \geq 0}_{\triangleq -g(\boldsymbol{\theta},\gamma,\boldsymbol{\eta})}, \tag{5}$$

where the sample-based approximation on the dataset $\mathcal{D}_{\text{IRL}}$ is used for the computation of the Wasserstein distance $f(\cdot)$, as well as for the constraint $g(\cdot)$.

Solving problems as (5) could be extremely challenging in the non-convex setting, where there are no widely-accepted optimization algorithms. Here we look at the min-max optimization as a competitive game between two players and seek for a stationary solution of the problem. Specifically, we reformulate (5) via the potential function $F(\boldsymbol{\theta},\gamma) \triangleq \max_{\boldsymbol{\eta}: g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0} f(\boldsymbol{\eta})$, obtaining $\min_{\boldsymbol{\theta}\in\mathbb{R}^{d_\theta},\,\gamma\in[0,1)} F(\boldsymbol{\theta},\gamma)$. If we assumed the concavity of $f(\cdot)$, we could compute, following [13], the gradient of $F(\cdot)$ as $\boldsymbol{\nabla}_{\boldsymbol{\theta},\gamma} F(\boldsymbol{\theta},\gamma) = \boldsymbol{\nabla}_{\boldsymbol{\theta},\gamma} f(\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma))$, where $\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma) = \arg\max_{\boldsymbol{\eta}: g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0} f(\boldsymbol{\eta})$, and reach a stationary point of (5). Some caution should be exercised here, since $\boldsymbol{\eta}^\star$ is an implicit function of $(\boldsymbol{\theta},\gamma)$, as it is defined by the constraint $g(\cdot)$. In place of partial derivatives, we should then resort to the following total differential forms:

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\theta}} = \frac{\partial F}{\partial \boldsymbol{\eta}}\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{\theta}}, \quad \text{with} \quad \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\boldsymbol{\theta}} = -\frac{\partial g}{\partial \boldsymbol{\theta}}\Big/\frac{\partial g}{\partial \boldsymbol{\eta}}, \qquad\qquad \frac{\mathrm{d}F}{\mathrm{d}\gamma} = \frac{\partial F}{\partial \boldsymbol{\eta}}\frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\gamma}, \quad \text{with} \quad \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\gamma} = -\frac{\partial g}{\partial \gamma}\Big/\frac{\partial g}{\partial \boldsymbol{\eta}}, \tag{6}$$

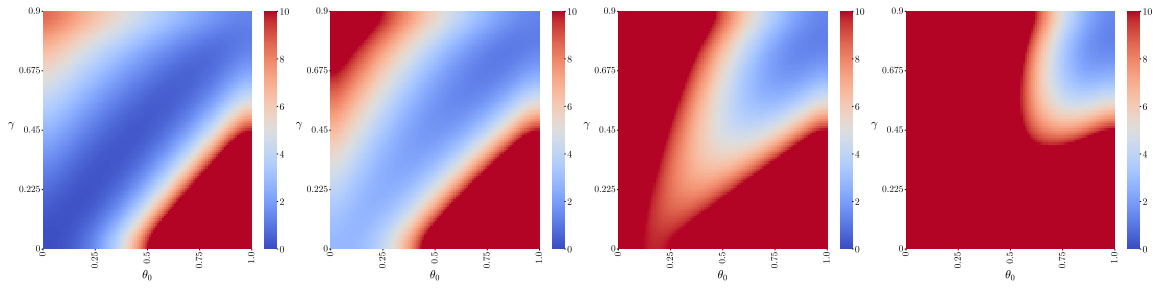
where the differentials and the divisions are to be intended component-wise, and the differentials of $\boldsymbol{\eta}$ are computed by applying to $g(\cdot) = 0$ the Implicit Function Theorem [6]. This iterative procedure would require to find the exact maximum solution of $\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma)$, which can be computationally unfeasible if the function $f(\cdot)$ is not concave. Fortunately, we can substitute $\boldsymbol{\eta}^\star$ with an approximate value $\widetilde{\boldsymbol{\eta}}^\star$ so as to satisfy the condition $f(\widetilde{\boldsymbol{\eta}}^\star) \geq \max_{\boldsymbol{\eta}: g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0} f(\boldsymbol{\eta}) - \epsilon$, and relax the concavity assumption. In this case, the algorithm is guaranteed [13] to find an approximate stationary point, where the accuracy level is given by the value of ϵ.

6 Experiments

As a proof of concept of the behaviour of our new IRL formulation, we run a set of experiments and investigate the main characteristics of the approach in the well-know Linear Quadratic Gaussian (LQG [3]) control problem. We consider the scalar case with nominal parameters, and compute in closed-form the expert policy π_E which is optimal for the reward $r_E(s,a) = -s^2 - a^2$ and $\gamma_E = 0.9$. The Q-function feature vector is $\boldsymbol{\psi} = [s^2, a^2, sa]$ so as to span the space of the exact Q-function $Q_{r_E,\gamma_E}^{\pi_E}$, while the reward features are set to $\boldsymbol{\phi} = [-s^2 - a^2, Q_{\bar{s}}^{\pi_E}(s,a)]$, where $Q_{\bar{s}}^{\pi_E}$ represent the Q-function of the expert in a shifted LQG problem with the goal in $\bar{s}$. In the following experiments, the policy is parametrized linearly in the state as $\pi_\eta(s) = \eta s$, and the reward weights $\boldsymbol{\theta}$ are normalized to sum to 1. Finally, we assumed to have an infinite number of samples to solve the forward learning problem, and set $M = \infty$.[2].

[2]We assumed to have a sufficiently high number of samples in the forward learning phase to reach the asymptotic behaviour $\epsilon_M \to 0$

For a complete numerical analysis of the new min-max formulation, we show in Figure 1 the values of the maximum Wasserstein distance $f(\boldsymbol{\eta}^\star)$ in (5) related to the change of the discount factor γ and the weights $\boldsymbol{\theta}$ of the reward $r_{\boldsymbol{\theta}}$, when we gradually increase the value of the goal state $\bar{s}$. As expected, when $\bar{s} = 0$, the formulation selects as the optimal min-max solution $\gamma^\star = 0$, thus minimizing the sample complexity for the forward learning phase, and $r_{\boldsymbol{\theta}^\star} = Q_{\bar{s}}^{\pi_E}$, which recovers the same behaviour of the expert's reward r_E. Interestingly, while the goal $\bar{s}$ moves from 0 (the expert's goal) to an higher value, the formulation trades off the sample complexity induced by a higher γ with the error induced on the learned policy when choosing a sub-optimal reward, moving towards the selection of the unbiased expert reward, selecting $\gamma^\star = \gamma_E$ and $r_{\boldsymbol{\theta}^\star} = r_E$ when the goal $\bar{s} = 0.4$ is too different (sub-optimal) w.r.t the expert goal 0.



(a) $\bar{s} = 0.0$ (b) $\bar{s} = 0.1$ (c) $\bar{s} = 0.2$ (d) $\bar{s} = 0.4$

Figure 1: Value of the objective function $f(\boldsymbol{\eta}^\star)$ in (5) related to the change of the outer variables $(\gamma, \boldsymbol{\theta})$, with $N = 200, c_N = 0.01$ and $M = \infty$. Each plot refers to different values of the goal $\bar{s}$.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings, Twenty-First International Conference on Machine Learning (ICML)*, pages 1–8, 2004.

[2] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91(3):325–349, 2013.

[3] Peter Dorato, Vito Cerone, and Chaouki Abdallah. *Linear-quadratic control: an introduction*. Simon & Schuster, Inc., 1994.

[4] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 4565–4573, 2016.

[5] Jonathan Ho, Jayesh K. Gupta, and Stefano Ermon. Model-free imitation learning with policy optimization. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 2760–2769, 2016.

[6] Steven G Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2012.

[7] Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003.

[8] Alessandro Lazaric, Mohammad Ghavamzadeh, and Remi Munos. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13:3041–3074, 2012.

[9] Alberto Maria Metelli, Matteo Pirotta, and Marcello Restelli. Compatible reward inverse reinforcement learning. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 2050–2059, 2017.

[10] Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 295–302, 2007.

[11] Andrew Y. Ng and Stuart J. Russell. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 663–670. Morgan Kaufmann Publishers Inc., 2000.

[12] Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2010.

[13] Meisam Razaviyayn, Tianjian Huang, Songtao Lu, Maher Nouiehed, Maziar Sanjabi, and Mingyi Hong. Non-convex min-max optimization: Applications, challenges, and recent theoretical advances. *arXiv:2006.08141*, Aug 2020. arXiv: 2006.08141.

[14] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, pages 1889–1897, 2015.

[15] Umar Syed, Michael Bowling, and Robert E. Schapire. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning (ICML)*, page 1032–1039. ACM Press, 2008.

A Proof of Proposition 4.2

The idea of this proposition is to start from the constraint (2)

$$\sum_{s\in\mathcal{S}} \widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_M^{\pi_E}(s,\pi_E(s)) \geq 0, \tag{7}$$

which includes the unknown quantity $\widehat{Q}_M^{\pi_E}$, and to compute a new looser inequality that involves only the known quantity $\widehat{Q}_N^{\pi_E}$. To make the above constraint looser, we need to take a larger LHS, *i.e.*, we need to consider an upper bound to $\widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))$ and a lower bound to $\widehat{Q}_M^{\pi_E}(s,\pi_E(s))$. Starting with the former, we can write:

$$\widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) = \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) + \left(\widehat{Q}\pi_E(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))\right) + \left(\widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))\right) \tag{8}$$

$$\leq \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) + \left|\widehat{Q}\pi_E(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))\right| + \left|\widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))\right| \tag{9}$$

$$\leq \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) + \epsilon_N + \epsilon_M, \tag{10}$$

where in the last step we applied the assumptions (3). Similarly, we can proceed with latter term, and derive

$$\widehat{Q}_M^{\pi_E}(s,\pi_E(s)) = \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) + \left(\widehat{Q}\pi_E(s,\pi_E(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_E(s))\right) + \left(\widehat{Q}_M^{\pi_E}(s,\pi_E(s)) - \widehat{Q}^{\pi_E}(s,\pi_E(s))\right) \tag{11}$$

$$\geq \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) - \left|\widehat{Q}\pi_E(s,\pi_E(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_E(s))\right| - \left|\widehat{Q}_M^{\pi_E}(s,\pi_E(s)) - \widehat{Q}^{\pi_E}(s,\pi_E(s))\right| \tag{12}$$

$$\geq \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) - \epsilon_N - \epsilon_M, \tag{13}$$

where again in the last step we applied the assumptions (3). Putting back together the computed upper and lower bound we obtain:

$$\sum_{s\in\mathcal{S}} \widehat{Q}_M^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_M^{\pi_E}(s,\pi_E(s)) \leq \sum_{s\in\mathcal{S}} \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) + 2\epsilon_N + 2\epsilon_M. \tag{14}$$

If the original constraint is satisfied, also the looser one holds.

B LQ case with a limited amount of samples

In order to highlight the effect of employing a possibly suboptimal reward when the forward RL phase has a limited number of samples at disposal, we design an additional experiment in the LQ setting. Specifically, we consider the two reward functions learned in the previous experiments $-s^2 - a^2$ and $Q_0^{\pi_E}(s,a)$. In Figure 2 we plot the learned parameter (top row) and the average discounted return (bottom row), when performing RL with REINFORCE in two different LQ environments. On the left, we consider the very same environment in which we performed the IRL phase, while on the right we consider an LQ in which we change the dynamical matrix (multiplied by 0.85 compared to the original setting). Thus, on the left, as expected, we observe that both reward functions $Q_0^{\pi_E}(s,a)$ (Controller with IRL reward) and $-s^2 - a^2$ (Controller with Real reward) are able to recover the optimal parameter, although $Q_0^{\pi_E}(s,a)$ requires a smaller number of samples. However, the interesting behavior is displayed on the right. While the original reward function $-s^2 - a^2$ is able to recover the correct parameter, when the number of samples is limited the *biased* reward $Q_0^{\pi_E}(s,a)$ learned in a different environment is more effective to achieve a reasonable performance. Clearly, as the number of samples increase the effect of bias is more visible.

The effect of using the optimized IRL reward on the sample complexity of the forward learning problem is also depicted in Figure 3. After solving the problem (5) (with parameters $N = 200, c_N = 0.01, M = \infty, \bar{s} = 0$), we employ the final pair $(\gamma^\star, \boldsymbol{\theta}^\star)$ to learn the optimal policy parameter as the number of available samples vary: in particular, we select 20 uniformly random initial states and then estimate the gradient direction in the REINFORCE algorithm by a Monte Carlo evaluation of the reward along trajectories of different length $H = \{1, 2, 6, 10\}$. The plot clearly shows how the IRL reward and discount factor allow the RL algorithm to reach the optimal value of the policy parameters much faster then using the LQG reward, *i.e.*, the number of samples processed during the learning process is much lower if the solution of our proposed IRL formulation is used in place of the expert's (and exact) one (γ_E, r_E).

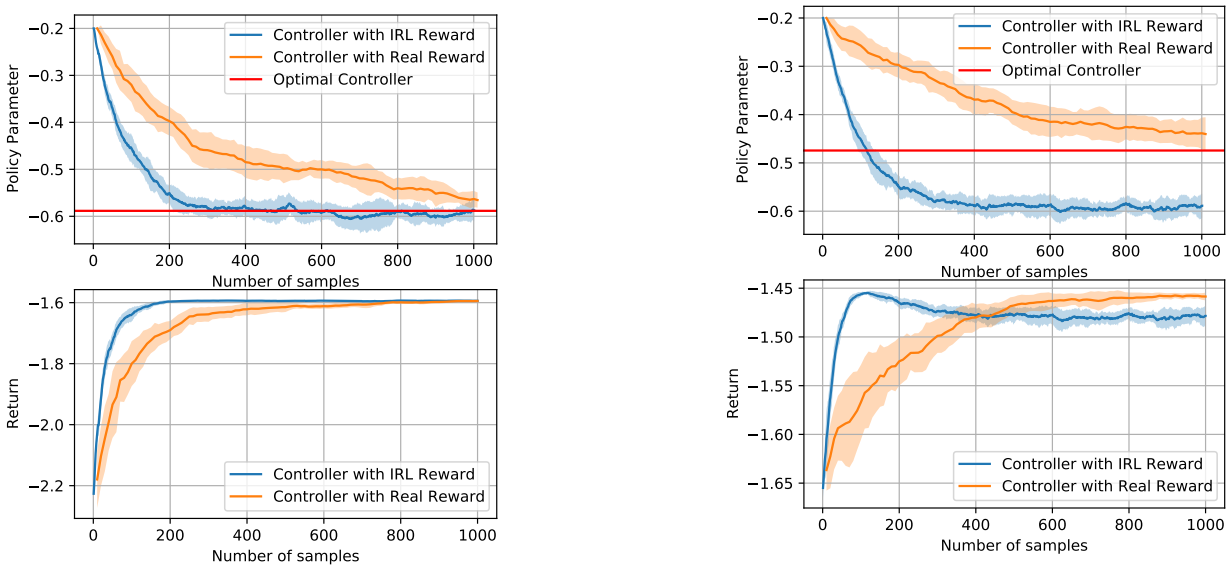


Figure 2: Comparison of the learned policy parameter and average return when learning in the same environment used for IRL (left) and when changing the environment (right) (10 runs, 95% c.i.).

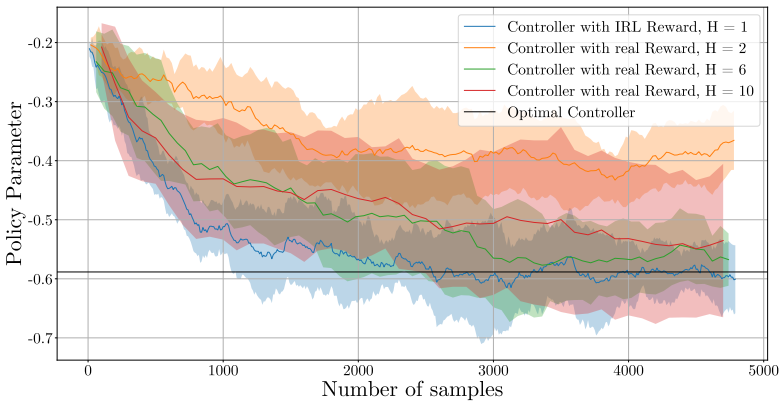


Figure 3: Impact of the optimized IRL reward on the sample efficiency of the forward learning task, and convergence to the expert's policy parameter (10 runs, 95% c.i.).