Balancing Sample Efficiency and Suboptimality in Inverse Reinforcement Learning

Angelo Damiani [*1] Giorgio Manganini [*1] Alberto Maria Metelli [*2] Marcello Restelli [2]

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 illustrative numerical cases.

1. Introduction

Inverse Reinforcement Learning (IRL, Ng & Russell, 2000) 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 behavior of the expert. The learned reward is intended to be successively used in forward Reinforcement Learning (RL, Sutton & Barto, 2018) to find new policies that could generalize over unseen states or even improve the expert's actions in new environments.

Learning policies from demonstrated examples, also known as Imitation Learning (Osa et al., 2018), is indeed often a much easier way to capture the expert's behavior compared to specifying an informative reward function, especially in complex problems (Ng & Russell, 2000), which usually involves some not straightforward tuning and tweaking to elicit the desired result. A direct approach to imitation learning, known as Behavioral Cloning (Bain & Sammut, 1995), involves the direct extrapolation of the policy function from the expert's demonstrations by using some supervised learning technique, but this may lead to surprisingly poor behaviors, especially under environment modifications (Ng & Russell, 2000; Ross & Bagnell, 2010; Bagnell, 2015). On the contrary, IRL methods are some of the most successful approaches to Imitation Learning, where first a reward function is learnt under the assumption of the expert's optimality, and then used to recover the expert's policy using forward RL (rather than mimic it). This accounts for a better generalization and transferability of the expert's intention, which is compactly described by its reward function.

Although theoretically sound, the IRL problem is scattered itself by a number of difficulties, to which many studies were devoted in the literature. The main dilemma is the "ill-posedness" of IRL, i.e., the fact that a policy can be optimal for multiple reward functions (Abbeel & Ng, 2004; Lopes et al., 2009). To obtain a unique solution, some approaches propose ad-hoc objective functions to be optimized, such as margin between the optimal policy and others (Ng & Russell, 2000; Abbeel & Ng, 2004; Ratliff et al., 2006; 2009; Silver et al., 2010), or the selection of a maximal entropy policy (Ziebart et al., 2008; Ziebart, 2010; Kitani et al., 2012; Shiarlis et al., 2016).

[*] Equal contribution [1] Department of Computer Science, Gran Sasso Science Institute, L'Aquila, Italy [2] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy. Correspondence to: Giorgio Manganini <giorgio.manganini@gssi.it>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

In their quest for the expert's reward function, many IRL approaches implement an iterative learning process (Abbeel & Ng, 2004; Syed et al., 2008a; Neu & Szepesvári, 2007; Ho & Ermon, 2016; Ho et al., 2016) that proceeds by alternately 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 must be enforced during the optimization of the reward function. Additionally, the learning of a generalized policy is then updated using a forward RL procedure based on the current estimate of the reward function.

A final common aspect to take care of in IRL approaches is the assumption about the underlying MDP. Traditionally, most IRL methods rely on model knowledge (either given or accurately learned from the demonstrated trajectories) (Ng & Russell, 2000; Ratliff et al., 2006; Ramachandran & Amir, 2007; Neu & Szepesvári, 2007; Syed et al., 2008a; Ziebart et al., 2008; Herman et al., 2016), 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 (Boularias et al., 2011; Ho et al., 2016; Pirotta & Restelli, 2016; Metelli et al., 2017; 2020; Ramponi et al., 2020; Likmeta et al., 2021), even though some of them still require continuous interactions with the environment (Abbeel & Ng, 2004; Ho & Ermon, 2016; Ho et al., 2016).

In this paper, we take a somewhat different point of view on IRL and focus our effort 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. All the methods presented above break the ambiguity by enforcing some particular property on the reward function directly (e.g., maximum entropy, maximum margin). However, they do not explicitly take into account *how* the recovered reward function will be employed. Indeed, such a reward will be plugged into (a possibly different) environment and used to perform forward RL. In this spirit, among the compatible ones, we should prefer the rewards that can 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. Our novel formulation 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 evaluation, and iii) does not require solving any forward problem (i.e., finding an optimal policy given a candidate reward function).

2. Preliminaries

Notation We denote with $\|f\|_\mu^2 = \int f(x)^2\mu(\mathrm{d}x)$ the square of the L_2-norm of f weighted by the probability distribution μ, and with $\|f\|_\infty = \sup_x f(x)$ the L_∞-norm of f. The L_2-Wasserstein distance between two probability distributions μ and ρ is defined as: $W_2^2(\mu, \rho) = \inf_{\xi \in \Xi(\mu,\rho)} \int \|x-y\|^2 \xi(\mathrm{d}x, \mathrm{d}y)$, where $\Xi(\mu, \rho)$ is the set of couplings with marginals μ and ρ, and $\|\cdot\|$ is some norm. Let $\mathcal{F}$ be a functional space and $\mathrm{proj}_{\mathcal{F}}\, g = \arg\min_{f \in \mathcal{F}} \|f-g\|$ the projection operator, where $\|\cdot\|$ is some norm.

Markov Decision Processes A Markov Decision Process (MDP, Puterman, 2014) is defined 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 Markovian state transition density $P(s'|s, a)$ defined for every triple (s', a, s), $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A Markovian stationary stochastic policy $\pi : \mathcal{A} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ specifies the action density function $\pi(a|s)$ defined for every pair (a, s). A policy is deterministic if, for any s, $\pi(\cdot|s)$ is concentrated on a single action. The class of Markovian stationary stochastic policies will be denoted by Π.

Value Functions, Operators, and Optimal Policy Given a reward function r and a discount factor γ, the state-action value function $Q^\pi_{r,\gamma} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as $Q^\pi_{r,\gamma}(s, a) \triangleq \mathbb{E}_\pi[\sum_{t=0}^{+\infty} \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a]$, where we denote with $\mathbb{E}_\pi$ the expected value w.r.t. the probability measure induced by executing π on $\mathcal{M}$. The state-action value function is the unique fixed-point of the Bellman operator $T^\pi : Q^\pi_{r,\gamma}(s, a) \mapsto r(s, a) + \gamma \int_{\mathcal{S}\times\mathcal{A}} \pi(a'|s')P(s'|s, a)Q^\pi(s', a')\mathrm{d}s'\mathrm{d}a'$. The optimal state–action value function is defined as $Q^\star_{r,\gamma}(s, a) \triangleq \max_{\pi\in\Pi} Q^\pi_{r,\gamma}(s, a)$ for every pair (s, a) (Puterman, 2014). Any policy $\pi \in \Pi$ satisfying the equality $\int_{\mathcal{A}} \pi(a|s)Q(s, a)\mathrm{d}a = \max_{a\in\mathcal{A}} Q(s, a)$ for all states $s \in \mathcal{S}$ is named *greedy* w.r.t. the function Q. We denote with $\mathcal{G}[Q] \subseteq \Pi$ the set of greedy policies w.r.t. Q. Any policy that is greedy w.r.t. the optimal state-action value function $Q^\star_{r,\gamma}$ is an *optimal policy* and is denoted by $\pi^\star_{r,\gamma}$.

Inverse Reinforcement Learning The IRL problem is informally defined as finding a reward function r that can explain the observed behavior of an expert following a policy π_E, which is optimal w.r.t. some unknown reward r_E (see Ng & Russell (2000)). Formally, a reward r is *compatible* with the expert's policy π_E if $\pi_E \in \Pi$ is optimal under r, i.e., $\pi_E \in \mathcal{G}[Q^\star_{r,\gamma}]$. Sometimes, especially in the Apprenticeship Learning class of algorithms, a learner policy π is obtained as a by-product of the IRL process. In these cases, the learner must find a policy that performs at least as well as the expert π_E on the unknown true reward function (Abbeel & Ng, 2004; Ramachandran & Amir, 2007; Neu & Szepesvári, 2007; Syed et al., 2008a).

3. The IRL Formulation for Efficient Forward Learning

In this section, we introduce our new IRL formulation and show how it is possible to select a reward that makes the forward RL task as efficient as possible. In doing so, we still need to guarantee that the derived reward is compatible with the expert demonstrations, i.e., that the corresponding optimal policy π matches the expert's policy π_E.

The issue of efficient learning is inextricably related to the concept of the sample complexity of finding a good approximation of the optimal value function and/or policy and, loosely speaking, can be questioned as *how much data must we collect in order to achieve "learning"?* (Kakade, 2003). In RL, the number of calls to the sampling model is generally a function of different problem parameters and, in particular, of the discount factor γ (linked to the effective number of decision epochs), and (the variance of) the reward, together with an accuracy parameter ϵ (w.r.t. the performance criteria used) and a confidence parameter δ. 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 witnessed by numerous sample complexity theoretical results, usually depending on a power of $1/(1-\gamma)$ (e.g., Munos & Szepesvári, 2008; Farahmand et al., 2010; Lazaric et al., 2012; Azar et al., 2013). Nevertheless, the reward function recovered by IRL has to be compatible with the expert's policy, which is known only with some accuracy and confidence, being estimated from a finite set of demonstrations.

The following novel IRL formulation blends all these different 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 that, provided with a reward function r, a discount factor γ, and a number of samples $M \geq 0$, is able to output an $\epsilon^\star(M,\gamma)$-approximation $\widehat{Q}^\star_M$ of the optimal Q-function $Q^\star_{r,\gamma}$, with probability at least $1-\delta^\star$. Then, the influence of the IRL reward r and discount factor γ on the distance between the expert's policy π_E and the learned policy in the successive forward learning task, when a finite and possibly small number M of samples is available, can be captured by the next adversarial min-max optimization program:

$$\min_{r\in\mathcal{R},\gamma\in[0,1)} \max_{\pi\in\mathcal{G}[\widehat{Q}^\star_M]} \left\| Q^{\pi_E}_{r_E,\gamma_E} - Q^{\pi}_{r_E,\gamma_E} \right\| \tag{1a}$$

$$\text{s.t.} \left\| \widehat{Q}^\star_M - Q^\star_{r,\gamma} \right\| \leq \epsilon^\star(M,\gamma), \tag{1b}$$

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 with respect to 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 π_E, 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 future estimated optimal Q-function $\widehat{Q}^\star_M$ under the optimized reward r and discount factor γ, which is determined by the used forward RL algorithm. 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 $\epsilon^\star$ and confidence level $\delta^\star$ varying with the number of data M available during the successive forward learning phase.

In summary, our novel IRL formulation (1) accounts for a new aspect that has not been taken into account by previous IRL approaches, i.e., the trade-off between the optimality (in the IRL sense, i.e., compatibility) and the sample complexity for learning a policy in the subsequent forward RL phase. The recovered reward aims at minimizing the worst-case error of the forward RL (objective (1a)) when a finite-sample budget M will be available in the forward RL phase (constraint (1b)). This does not imply that the policy will be easy to learn in general, but the optimized pair (r,γ)[1] will make the forward RL policy as efficient as possible to learn.

We finally remark that formulation (1) is not readily solvable, because it involves the unknown quantities r_E and γ_E in its objective function (1a). In Section 4, we will show how we can get rid of the state-action value functions $Q^{\pi_E}_{r_E,\gamma_E}$ and $Q^{\pi}_{r_E,\gamma_E}$, and how we can determine an approximation of the confidence region in (1b). The numerical scheme for solving (1) will be described in Section 5.

4. Construction of a Solvable IRL Formulation

We devote this section to the description of the main elements involved in the formulation (1), and to the construction of a numerically solvable optimization problem.

Parametrizations First, we introduce some widely accepted approximations, and assume that there are vectors of 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 can linearly parameterize the reward function and the action-value function, such that $r_{\boldsymbol{\theta}}(s,a) = \boldsymbol{\phi}(s,a)^\top\boldsymbol{\theta}$ and $\widehat{Q}_{\boldsymbol{\omega}}(s,a;\boldsymbol{\omega}) = \boldsymbol{\psi}(s,a)^\top\boldsymbol{\omega}$, where $\boldsymbol{\theta}\in\mathbb{R}^{d_\theta}$ and $\boldsymbol{\omega}\in\mathbb{R}^{d_\omega}$.[2] Also, we restrict the search

[1] (r,γ) are reported under the min operator because they are the optimization variables that influence the constraint (1b) and hence the objective function in an indirect way.

[2] We require the features $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ to be linearly independent

of the greedy policy in the inner maximization over a class of parametrized policies $\Pi_{\boldsymbol{\eta}} = \left\{\pi_{\boldsymbol{\eta}} : \boldsymbol{\eta} \in \mathbb{R}^{d_\eta}\right\}$, with the only constraint that $\pi_{\boldsymbol{\eta}}$ is differentiable w.r.t. $\boldsymbol{\eta}$.

Relaxation Steps Second, we then proceed with the manipulation of the objective function and the constraint of the formulation (1) according to the following steps.

(i) Section 4.1: substitution in (1a) of the distance between Q-functions with the distance between the policies π and π_E, so as to avoid the usage of the unknown quantities r_E and γ_E;

(ii) Section 4.2: replacement of the forward optimal Q-function $Q^\star_{r,\gamma}$ in the constraint (1b), since it is not available during the IRL problem, with one computable during the IRL task, say $\widehat{Q}^{\pi_E}_N$, assuming to have access to a budget of N samples during IRL (instead of M);

(iii) Section 4.3: estimation of the IRL Q-function $\widehat{Q}^{\pi_E}_N$ and definition of its confidence interval.

4.1. Wasserstein Distance on Expert's Policy π_E

The formulation of the optimization problem (1) cannot be solved directly as it involves in the objective the expert's reward r_E and discount factor γ_E, which are clearly unknown. Thus, we consider a surrogate objective function, and bypass the value-function distance by introducing a policy divergence. Specifically, the following result, due to (Rachelson & Lagoudakis, 2010; Pirotta et al., 2015), bounds the Q-function distance with the policy distance.

Theorem 4.1 (Rachelson & Lagoudakis (2010))**.** *If the MDP is Lipschitz continuous with constants (L_r, L_P) and π is Lipschitz continuous with constant L_π. Then, it holds that:*

$$\left\| Q^{\pi_E}_{r_E,\gamma_E} - Q^{\pi}_{r_E,\gamma_E} \right\|_\mu \le L_Q \mathop{\mathbb{E}}_{s \sim d^{\pi_E}_{\mu,\gamma_E}} \left[W_2(\pi_E(\cdot|s), \pi(\cdot|s)) \right],$$

where $L_Q = \frac{\gamma_E L_r L_\pi}{(1-\gamma_E)(1-\gamma_E L_P(1+L_\pi))}$, W_2 is the L_2-Wasserstein distance and $d^{\pi_E}_{\mu,\gamma_E}$ is the γ_E-discounted state occupancy (Sutton et al., 1999) induced by policy π_E and initial state distribution μ.

Since we aim to deal with continuous action spaces and deterministic policies (the expert's policy is usually deterministic), the Wasserstein's distance (Villani, 2008) is an appropriate distributional divergence.[3] Formally, given two deterministic policies $\pi_{\boldsymbol{\eta}}$ and π_E, and a state $s \in \mathcal{S}$, we can compute the L_2-Wasserstein distance as:[4]

$$W_2^2(\pi_E(s), \pi_{\boldsymbol{\eta}}(s)) = (\pi_E(s) - \pi_{\boldsymbol{\eta}}(s))^2. \qquad (2)$$

Remark 4.2*.* Thanks to Theorem 4.1, we can remove from the formulation (1) the dependence on expert's reward r_E and discount factor γ_E. From now on, unless explicitly stated, all the Q-functions will refer to the optimized pair (r, γ), which, for sake of compactness, will be removed from the subscripts.

4.2. Dealing with the Forward Q-function $Q^\star_{r,\gamma}$

Removing, in the previous sections, the dependence on the expert's reward and discount factor from the definition of the objective function (1a) is not sufficient yet to translate the program (1) into a solvable one. The main open issue is that the confidence region of the constraint (1b) involves the unknown quantity $Q^\star_{r,\gamma}$, i.e., the optimal Q-function computed with the optimized pair (r, γ).

Replacing $Q^\star_{r,\gamma}$ with $Q^{\pi_E}_{r,\gamma}$ In principle, one could execute forward RL as an inner loop for every candidate pair (r, γ) and come up with an approximation of $Q^\star_{r,\gamma}$ and employ it to enforce the constraint (1b), at least in an approximate way. However, this choice is not viable in our setting, as we have access to just the budget of N samples available at IRL time and, in any case, we want to avoid performing forward RL. To this end, we replace in constraint (1b) the optimal Q-function $Q^\star_{r,\gamma}$ with the expert Q-function $Q^{\pi_E}_{r,\gamma}$. The rationale behind this approximation is that, when (r, γ) are compatible with the expert's policy π_E then it holds that $Q^\star_{r,\gamma} = Q^{\pi_E}_{r,\gamma}$. This approximation has the remarkable advantage of not requiring to perform forward RL, but just *policy evaluation*. Indeed, we need to estimate, for every candidate pair (r, γ), the expert's Q-function $Q^{\pi_E}_{r,\gamma}$ only. Thus, constraint (1b) is substituted with:

$$\left\| \widehat{Q}^{\pi_E}_M - Q^{\pi_E}_{r,\gamma} \right\| \le \epsilon_1(M, \gamma), \qquad (3)$$

where the $\epsilon_1(M, \gamma)$ depends on the policy evaluation algorithm employed to estimate $\widehat{Q}^{\pi_E}_M$ and the bound holds with probability at least $1 - \delta_1$. The policy evaluation task will be easily performed leveraging the N samples available at IRL time.

Relaxing the Greedy Constraint A second issue concerns the computation of the greedy policy to perform the inner maximization. To counteract this obstacle, we begin by rewriting explicitly the greedy constraint on the policy in the inner maximization of (1a) and performing two relaxations:

to ensure that there are no redundant parameters and that the subsequent computations involve full-rank matrices.

[3] Other common divergences, like Total Variation or Kullback-Leibler are unsuited for deterministic distributions since they will provide the maximum distance value.

[4] To ease the exposition, we deal with deterministic policies, but the formulation and, in particular, the Wasserstein distance, can be extended to stochastic policies.

$$\begin{aligned}
&\pi_{\boldsymbol{\eta}} \in \mathcal{G}\left[\widehat{Q}^{\pi_E}_M\right] \\
&\Rightarrow \widehat{Q}^{\pi_E}_M(s, \pi_{\boldsymbol{\eta}}(s)) \geq \widehat{Q}^{\pi_E}_M(s, \pi_E(s)) \quad \forall s \in \mathcal{S} \qquad (4)\\
&\Rightarrow \sum_{s \in \mathcal{D}_{\text{IRL}}} \widehat{Q}^{\pi_E}_M(s, \pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_M(s, \pi_E(s)) \geq 0.
\end{aligned}$$

The first relaxation involves the transition from a greedy policy to all policies with at least a *performance improvement* (i.e., positive advantage), so as to have an explicit dependence of the learner policy $\pi_{\boldsymbol{\eta}}$ on $\widehat{Q}^{\pi_E}_M$. 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 to enforce it in an infinite state space (a similar relaxation is operated for instance in (Schulman et al., 2015) for the KL-divergence).

Enforcing Constraint (3) Finally, in order to enforce constraint (3), we show how to obtain a relaxed version in combination with the policy improvement inequality (4) to remove from problem (1) the dependence on the expert's Q-function $Q^{\pi_E}_{r,\gamma}$. The idea is to compute a looser constraint than (3) but which does not involve the unknown quantity $Q^{\pi_E}_{r,\gamma}$.

Proposition 4.3. *Suppose that, simultaneously for all $r \in \mathcal{R}$ and $\gamma \in [0, 1)$, we have that the Q-function $\widehat{Q}^{\pi_E}_M$ is known within some accuracy level $\epsilon_1(M, \gamma)$ (with probability $1 - \delta_1$), and let us introduce a new Q-function $\widehat{Q}^{\pi_E}_N$ which can be estimated with N samples during the IRL task within some accuracy $\epsilon_2(N, \gamma)$ (with probability $1 - \delta_2$), i.e.,:*

$$\left\|\widehat{Q}^{\pi_E}_M - Q^{\pi_E}_{r,\gamma}\right\|_\infty \leq \epsilon_1(M, \gamma), \qquad (5)$$

$$\left\|\widehat{Q}^{\pi_E}_N - Q^{\pi_E}_{r,\gamma}\right\|_\infty \leq \epsilon_2(N, \gamma). \qquad (6)$$

Then, with probability at least $1 - \delta_1 - \delta_2$, the inequality (4) can be upper bounded by dropping the dependence on $\widehat{Q}^{\pi_E}_M$, obtaining the following inequality:

$$\begin{aligned}
\sum_{s \in \mathcal{D}_{\text{IRL}}} & \widehat{Q}^{\pi_E}_N(s, \pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_N(s, \pi_E(s)) \\
& + 2\epsilon_1(M, \gamma) + 2\epsilon_2(N, \gamma) \geq 0. \qquad (7)
\end{aligned}$$

Proof. See Appendix A. □

As for the parameters $\epsilon_1(M, \gamma)$ and $\epsilon_2(N, \gamma)$, in the following we consider the general structures:

$$\epsilon_1(M,\gamma) = \frac{\gamma c_1}{(1-\gamma)\sqrt{M}}, \quad \epsilon_2(N,\gamma) = \frac{\gamma c_2}{(1-\gamma)\sqrt{N}}, \qquad (8)$$

which generalize most of the sample complexity bounds available in the literature, typically asymptotic to a power of $1/(1-\gamma)$ (e.g., Munos & Szepesvári, 2008; Farahmand et al., 2010; Lazaric et al., 2012; Azar et al., 2013). Parameters c_1 and c_2 are appropriate constant factors determined by specific finite-sample analysis, and they usually capture the characteristics of the underlying MDP, together with the choice of the particular estimation algorithm.

Remark 4.4. We remark that the assumption (5) is simply the constraint (3) itself instanced with the L_∞-norm, which we directly use here to build the new relation involving only $\widehat{Q}^{\pi_E}_N$. In particular, inequality (7) comprises all the uncertainties related either with $\widehat{Q}^{\pi_E}_M$ and $\widehat{Q}^{\pi_E}_N$, and it depends on both the outer optimization variables γ (via parameters ϵ_1 and ϵ_2, and the Q-function $\widehat{Q}^{\pi_E}_N$) and r (via the Q-function $\widehat{Q}^{\pi_E}_N$), as well as on the number of samples N available for the IRL task (rather than on the number of samples M that will be used in the forward RL problem). Dependence of (7) on the policy $\pi_{\boldsymbol{\eta}}$ is instead straightforward.

4.3. Expert's Policy Evaluation with $\widehat{Q}^{\pi_E}_N$

The final stage in our construction of a solvable IRL formulation is the estimation of the new Q-function $\widehat{Q}^{\pi_E}_N$. While, in principle, any policy evaluation algorithm may be used to this purpose, here we resort to the the Least-Squares Temporal Difference (LSTDQ, Lagoudakis & Parr, 2003) algorithm, for which a confidence region of the form (6) is available via finite-sample analysis. In summary, LSTDQ returns the parameter vector $\boldsymbol{\omega} \in \mathbb{R}^{d_\omega}$ that minimizes the mean-squread projected Bellman error (MSPBE):

$$\text{MSPBE}(\boldsymbol{\omega}) \triangleq \left\|\widehat{Q}_{\boldsymbol{\omega}} - \text{proj}_\Psi T^{\pi_E}\widehat{Q}_{\boldsymbol{\omega}}\right\|^2_{\rho^{\pi_E}}, \qquad (9)$$

where proj_Ψ is the projection operator onto the linear space Ψ spanned by the basis functions ψ and ρ^{π_E} is the stationary distribution induced by π_E. Due to the employed linear parametrization, the projection operator turns out to be linear as well and it is given by the matrix representation $\boldsymbol{\Psi}(\boldsymbol{\Psi}^\top\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^\top$, where $\boldsymbol{\Psi} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d_\omega}$ is a matrix with feature vectors $\boldsymbol{\psi}(s, a)^\top$ as rows for every $(s, a) \in \mathcal{S} \times \mathcal{A}$.[5]

LSTDQ relies on a batch of transitions (even collected off-policy) $\mathcal{D}_{\text{LSTD}} = \{(s_i, a_i, s'_i, r_i)\}_{i=1}^N$, where $s'_i \sim P(\cdot|s_i, a_i)$ is the next state and $r_i = \boldsymbol{\phi}(s_i, a_i)^\top\boldsymbol{\theta}$ is the reward, to numerically evaluate the Bellman operator, thus avoiding the knowledge of the underlying MDP, and to sample the full feature matrix $\boldsymbol{\Psi}$, which cannot be formed in a continuous environment.[6] By defining the sample vectors $\tilde{\boldsymbol{\Psi}} \triangleq \left[\boldsymbol{\psi}(s_1, a_1)^\top; \ldots; \boldsymbol{\psi}(s_N, a_N)^\top\right]$, $\tilde{\boldsymbol{\Psi}}' \triangleq \left[\boldsymbol{\psi}(s'_1, \pi_E(s'_1))^\top; \ldots; \boldsymbol{\psi}(s'_N, \pi_E(s'_N))^\top\right]$, and

[5] With a slight abuse of notation, we overload the symbols Ψ and $\boldsymbol{\Psi}$ to represent either the space and the matrix of the basis functions ψ.

[6] In our IRL setting, the dataset $\mathcal{D}_{\text{LSTD}}$ can be the same used for solving the IRL, i.e., $\mathcal{D}_{\text{IRL}}$ or, if available, a newly collected dataset.

$\tilde{\boldsymbol{\Phi}} \triangleq \left[\boldsymbol{\phi}(s_1,a_1)^\top;\ldots;\boldsymbol{\phi}(s_N,a_N)^\top\right]$, the above equation can be manipulated to obtain the empirical MSPBE as a standard least-squares problem:

$$\widehat{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}} \left\|\boldsymbol{A}\boldsymbol{\omega} - \boldsymbol{b}\right\|^2 = \boldsymbol{A}^{-1}\boldsymbol{b}, \tag{10}$$

where $\boldsymbol{A} \triangleq \tilde{\boldsymbol{\Psi}}^\top(\tilde{\boldsymbol{\Psi}} - \gamma\tilde{\boldsymbol{\Psi}}')$, and $\boldsymbol{b} \triangleq \tilde{\boldsymbol{\Psi}}^\top(\tilde{\boldsymbol{\Phi}}\boldsymbol{\theta})$. It is worth noticing that the terms $\boldsymbol{A}$ and $\boldsymbol{b}$ depend upon the discount factor γ and the parameterized reward $\tilde{\boldsymbol{\Phi}}\boldsymbol{\theta}$ evaluated on the samples in $\mathcal{D}_{\text{LSTD}}$, respectively. Finally, it has been shown (Bradtke & Barto, 1996; Nedić & Bertsekas, 2003; Lazaric et al., 2012) that the LSTDQ solution $\tilde{\boldsymbol{\Psi}}\widehat{\boldsymbol{\omega}}$ converges to the fixed-point of $\operatorname{proj}_\Psi T^{\pi_E} Q^{\pi_E}$ as $N \to \infty$.

As anticipated above, the choice of LSTDQ allows us to provide a specific confidence region for $\widehat{Q}_N^{\pi_E}$ of the form (6). A useful result to this end is given by (Lazaric et al., 2012, Theorem 5), where the authors derived a finite-sample analysis of the LSTD algorithm. Here below we reformulate this theorem to provide a bound on the prediction error of the LSTDQ solution $\widehat{Q}_{\widehat{\boldsymbol{\omega}}} = \widehat{Q}_N^{\pi_E}(s,a;\widehat{\boldsymbol{\omega}})$ w.r.t. the true value function $Q_{r,\gamma}^{\pi_E}(s,a)$. The theorem holds under the assumption that, with probability $1-\delta$, the sample-based Gram matrix $\frac{1}{N}\tilde{\boldsymbol{\Psi}}^\top\tilde{\boldsymbol{\Psi}}$ is invertible and its smallest eigenvalue ν_N is positive, which is guaranteed if the number of samples N satisfies the condition (up to constant and logarithmic factors) $N \geq \tilde{\mathcal{O}}\Big(288L^2/\nu(d+1)\log(N/\delta)^2\Big)$, where ν is the smallest eigenvalue of the exact Gram matrix (see Lazaric et al., 2012, Lemma 4).

Theorem 4.5 (Lazaric et al. (2012))**.** *Let* (s_1,a_1), $\ldots,(s_N,a_N)$ *be a path generated by a stationary* β*-mixing process with parameters* $\bar{\beta}, b, \kappa$ *(that is, its* β*-mixing coefficients satisfy* $\beta_i \leq \bar{\beta}\exp(-bi^\kappa)$*) and stationary distribution* ρ^{π_E}*. With probability at least* $1-\delta_2$ *we have:*

$$\left\|Q_{r,\gamma}^{\pi_E} - \widehat{Q}_N^{\pi_E}\right\|_{\rho^{\pi_E}} \leq \epsilon_2(N,\gamma) = \tag{11}$$
$$\frac{2}{\sqrt{1-\gamma^2}}\left(2\sqrt{2}\left\|Q_{r,\gamma}^{\pi_E} - \operatorname{proj}_\Psi Q_{r,\gamma}^{\pi_E}\right\|_\rho + \epsilon^{(1)}\right) + \epsilon^{(2)}$$
$$+ \frac{2}{1-\gamma}\left[\gamma Q_{\max} L\sqrt{\frac{d}{\nu}}\left(\sqrt{\frac{8\log\left(8K/\delta_N\right)}{N}} + \frac{1}{N}\right)\right],$$

with $\epsilon^{(1)}$ *and* $\epsilon^{(2)}$ *that are* $\mathcal{O}(Q_{\max}/\sqrt{N})$*, where* $Q_{\max} = \|r\|_\infty/(1-\gamma)$*, and* L *is the upper bound of each basis function* ψ_j *of a feature vector* $\boldsymbol{\psi}$*, i.e.,* $\|\psi_j\|_\infty \leq L$*.*

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 optimization 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:

$$\min_{\substack{\boldsymbol{\theta}\in\mathbb{R}^{d_\theta}\\ \gamma\in[0,1)}} \max_{\boldsymbol{\eta}\in\mathbb{R}^{d_\eta}} \underbrace{\sum_{s\in\mathcal{D}_{\text{IRL}}} W_2(\pi_E(s),\pi_{\boldsymbol{\eta}}(s)),}_{\triangleq f(\boldsymbol{\eta})} \qquad \text{s.t.} \tag{12}$$
$$\underbrace{\sum_{s\in\mathcal{D}_{\text{IRL}}} \widehat{Q}_N^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}_N^{\pi_E}(s,\pi_E(s)) + 2\epsilon_M + 2\epsilon_N \geq 0,}_{\triangleq -g(\boldsymbol{\theta},\gamma,\boldsymbol{\eta})}$$

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 $g(\cdot)$ (the dependence of $g(\cdot)$ on the optimization variables is described in Remark 4.4).

When min-max problems are concave in the inner variables (i.e., $\boldsymbol{\eta}$) and convex in the outer variables (i.e., $\boldsymbol{\theta},\gamma$), a wide range of algorithms have been proposed in the literature (Wang & Li, 2020). On the contrary, solving problems as (12) could be extremely challenging in the non-convex setting, where there are no widely-accepted optimization algorithms. For instance, the naïve extension of a gradient-like descent-ascent algorithm to the min-max setting may easily fail to converge to any meaningful point (Razaviyayn et al., 2020).

Here we look at the min-max optimization as a competitive game between two players and seek for a stationary solution of the problem. Following (Razaviyayn et al., 2020), we reformulate (12) via the potential function $F(\boldsymbol{\theta},\gamma) \triangleq \max_{\boldsymbol{\eta}\in\mathbb{R}^{d_\eta}: g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0} f(\boldsymbol{\eta})$, obtaining:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{d_\theta},\,\gamma\in[0,1)} F(\boldsymbol{\theta},\gamma).$$

Under the assumption that there exists an implicit function $\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma) \triangleq \arg\max_{\boldsymbol{\eta}\in\mathbb{R}^{d_\eta}: g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0} f(\boldsymbol{\eta})$, we can compute the gradient of $F(\cdot)$ using the chain rule as follows: $\boldsymbol{\nabla}_{\boldsymbol{\theta},\gamma}F(\boldsymbol{\theta},\gamma) = \boldsymbol{\nabla}_{\boldsymbol{\theta},\gamma}f(\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma)) = \boldsymbol{\nabla}_{\boldsymbol{\theta},\gamma}\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma)\boldsymbol{\nabla}_{\boldsymbol{\eta}}f(\boldsymbol{\eta})|_{\boldsymbol{\eta}=\boldsymbol{\eta}^\star(\boldsymbol{\theta},\gamma)}$ (Danskin, 1966). Some caution should be exercised here, since $\boldsymbol{\eta}^\star$ is an implicit function of $(\boldsymbol{\theta},\gamma)$, as it is defined by satisfaction of the constraint $g(\cdot)=0$. In place of partial derivatives, we should then resort to the following choice of total differential forms:

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\theta}} = \frac{\partial f}{\partial\boldsymbol{\eta}^\star}\frac{\mathrm{d}\boldsymbol{\eta}^\star}{\mathrm{d}\boldsymbol{\theta}}, \quad \text{with} \quad \frac{\mathrm{d}\boldsymbol{\eta}^\star}{\mathrm{d}\boldsymbol{\theta}} = -\frac{\partial g}{\partial\boldsymbol{\theta}}\bigg/\frac{\partial g}{\partial\boldsymbol{\eta}^\star}, \tag{13a}$$
$$\frac{\mathrm{d}F}{\mathrm{d}\gamma} = \frac{\partial f}{\partial\boldsymbol{\eta}^\star}\frac{\mathrm{d}\boldsymbol{\eta}^\star}{\mathrm{d}\gamma}, \quad \text{with} \quad \frac{\mathrm{d}\boldsymbol{\eta}^\star}{\mathrm{d}\gamma} = -\frac{\partial g}{\partial\gamma}\bigg/\frac{\partial g}{\partial\boldsymbol{\eta}^\star}, \tag{13b}$$

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 (Krantz & Parks, 2012). We can now finally solve problem (12) by running the following iterative procedure, for $t \in \mathbb{N}$:

$$\boldsymbol{\eta}_{t+1} = \underset{\boldsymbol{\eta}\in\mathbb{R}^{d_\eta}: g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0}{\arg\max} \quad f(\boldsymbol{\eta}), \tag{14a}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_\theta \left.\frac{\mathrm{d}F(\boldsymbol{\eta}_{t+1},\boldsymbol{\theta},\gamma_t)}{\mathrm{d}\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t},$$

$$\gamma_{t+1} = \mathrm{proj}_{[0,1)}\left[\gamma_t - \alpha_\gamma \left.\frac{\mathrm{d}F(\boldsymbol{\eta}_{t+1},\boldsymbol{\theta}_t,\gamma)}{\mathrm{d}\gamma}\right|_{\gamma=\gamma_t}\right],$$

where $\alpha_\theta > 0$ and $\alpha_\gamma > 0$ are suitable learning rates.

The previous algorithm would require to find the exact solution of the maximization in (14a), which can be computationally unfeasible if the function $f(\cdot)$ is not concave. Fortunately, we can substitute an approximate computation of the point $\boldsymbol{\eta}_{r+1}$ so as to satisfy the condition $f(\boldsymbol{\eta}_{r+1}) \geq \max_{\boldsymbol{\eta}\in\mathbb{R}^{d_\eta}:g(\boldsymbol{\eta},\boldsymbol{\theta},\gamma)\leq 0} f(\boldsymbol{\eta}) - \epsilon'$, and relax the concavity assumption. In this case, the algorithm is guaranteed (Razaviyayn et al., 2020) to find an approximate stationary point (with suitable choice of α_θ and α_γ), where the accuracy level is given by the value of $\epsilon' > 0$.

6. Related work

Existing algorithms for IRL or Apprenticeship Learning have focused mainly on solving the issue of ill-posedness (Abbeel & Ng, 2004; Lopes et al., 2009) between the expert policy and multiple reward functions, and proposed methods to differentiate and rank different policies according to unique criteria. In particular, most of the approaches in the literature either use an algorithm for finding the MDP optimal policy (e.g., value iteration or policy iteration) as a subroutine in the IRL procedure, or return apprentice policies that are randomized combinations of stationary policies. Our min-max formulation takes a completely different point of view, and seeks a reward that results in an efficient forward learning problem, while still retaining the compatibility with the expert. Thanks to the max selection, the ambiguity on equivalent policies is avoided by construction. Furthermore, no planning algorithm is involved in our procedure.

Beyond these two main distinctions, the closest works in the literature to ours are those based on a game-theoretic view of the IRL problem. In (Syed et al., 2008a), a two-player zero-sum game is presented to select a policy that can perform significantly better than the expert's one by maximizing the worst-case performance in the case of a wrongly chosen reward. However, their algorithm requires the repeated solution of a planning problem. Moreover, the same authors, focusing on the computational complexity of the apprenticeship learning task, cast in (Syed et al., 2008b) their max-min problem as a linear program to improve the running time of their algorithm.

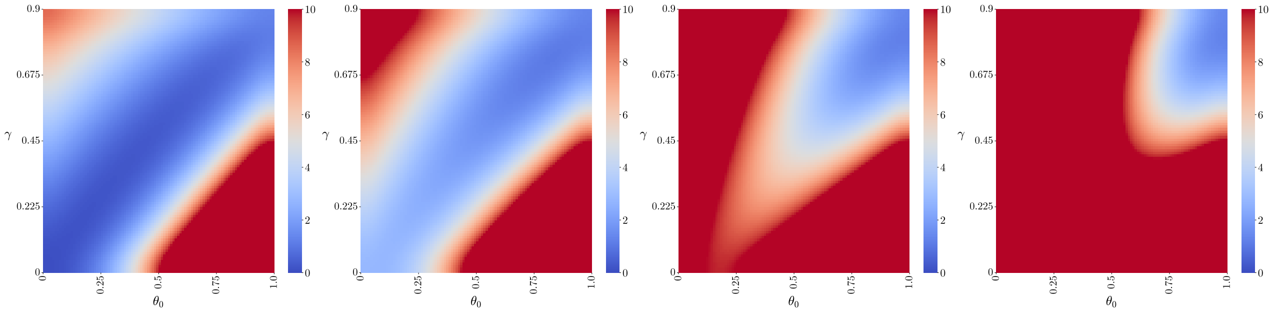
A different perspective is discussed in (Ho & Ermon, 2016), where the authors propose a new framework that directly derives a policy as if it were obtained by reinforcement learning following IRL, cast as a generative adversarial imitation learning problem. Despite the common attention for the forward learning task, the goal of (Ho & Ermon, 2016) is put on the sample efficiency of IRL in terms of expert data, whereas it requires several environment interactions during train. On the contrary, our formulation aims to find a sample-efficient reward for the forward learning task, whatever learning algorithm will be used, and, at the same time, it does not require any interaction with the environment during IRL.

Another min-max formulation is designed in (Ho et al., 2016) to determine a parametrized stochastic policy that performs at least as well as the expert's policy on an unknown reward function. This model-free algorithm avoids the explicit learning of the reward to remove the need to understand whether it is compatible with the expert or not, which is usually done by running an expensive reinforcement learning algorithm on it (Neu & Szepesvári, 2009). We reach the same objective by directly guaranteeing, in a worst-case scenario, the compatibility of the learned reward with the expert's policy, without accessing any environment but implicitly evaluating the performance of the learned policy over the expert data.

7. Numerical Simulations

As a proof of concept of the need for a reward function that is aware of the subsequent forward RL phase and to investigate the behavior our new IRL formulation, we run a set of experiments in Linear Quadratic Gaussian (LQG) control problem (Dorato et al., 1994) and in the Mountain Car domain (Moore, 1990).

Linear Quadratic Gaussian Regulator We consider a scalar LQG problem 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 $\boldsymbol{\psi}(s,a) = [s^2, a^2, sa]$ so as to span the space of the exact Q-function $Q^{\pi_E}_{r_E,\gamma_E}$, while the reward features are set to $\boldsymbol{\phi}(s,a) = [-s^2 - a^2, Q^{\pi_E}_{\bar{s}}(s,a)]$, where $Q^{\pi_E}_{\bar{s}}$ represent the Q-function of the expert in a shifted LQG problem with the goal in $\bar{s} \neq 0$ (i.e., the expert is optimal w.r.t. the reward $r_{\bar{s}}(s,a) = -(s-\bar{s})^2 - (a-\bar{a})^2$, with $\bar{a}$ being the equilibrium control action corresponding to $\bar{s}$). In all 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. The dataset $\mathcal{D}_{\mathrm{LSTD}}$ for the estimation of $\widehat{Q}^{\pi_E}_N$ has been generated starting from 40 uniformly sampled states in the interval $[-1, 1]$ and following for $H = 5$ steps the expert policy, whose actions were corrupted by a white noise with standard deviation of 0.05. The dataset $\mathcal{D}_{\mathrm{IRL}}$ for the resolution of the IRL formulation



(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 (12) related to the change of the outer variables $(\gamma, \boldsymbol{\theta})$, with $N = 200, c_1 = 0.01$ and $M = \infty$. Each plot refers to different values of the goal $\bar{s}$.

has been set to 200 randomly sampled states in the interval $[-1, 1]$, and hence $N = 200$. Finally, we assumed to have an infinite number of samples to solve the forward learning problem, and set $M = \infty$.[7]

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 (12) 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 $r_{\boldsymbol{\theta}^\star} = Q^{\pi_E}_{\bar{s}}$, which recovers the same behavior of the expert's reward r_E. Interestingly, while the goal $\bar{s}$ moves from 0 (the expert's goal) to a 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.

In order to highlight the effect of employing a possibly sub-optimal reward when the forward RL phase has a limited number of samples at disposal, we design an additional experiment in the LQG setting. Specifically, we consider the two reward functions learned in the previous experiments $-s^2 - a^2$ and $Q^{\pi_E}_0(s, a)$. In Figure 2, we plot the learned parameter (top row) and the average discounted return (bottom row), when performing RL with REINFORCE (Williams, 1992) in two different LQG environments. On the left, we consider the very same environment in which we performed the IRL phase, while on the right we consider an LQG 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 $Q^{\pi_E}_0(s, a)$ (Controller with IRL reward) and $-s^2 - a^2$ (Controller with Real reward) are able to recover the optimal parameter, although $Q^{\pi_E}_0(s, a)$ 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 $Q^{\pi_E}_0(s, a)$ learned in a different environment is more effective to achieve a reasonable performance. Clearly, as the number of samples increases 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 (12) (with parameters $N = 200, c_1 = 0.01, M = \infty, \bar{s} = 0$), we employ the final pair $(\gamma^\star, \boldsymbol{\theta}^\star)$ to learn the optimal policy parameter as the number of available samples varies. In particular, we select 20 uniformly random initial states and then estimate the gradient direction in the REINFORCE (Williams, 1992) algorithm by a Monte Carlo evaluation of the reward along trajectories of different lengths (we used $H = 1$ with the IRL reward and $H \in \{2, 6, 10\}$ with the real one). 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 than 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).

Mountain Car To further support the need for a reward function that is aware of the budget available in the forward RL phase, we provide an additional experiment in the Mountain Car domain (Moore, 1990). We consider two different reward functions:

(i) $r_0(s, a) = 200 \cdot \mathbf{1}_{[s=\text{goal}]}(s) - a^2$: an almost *sparse* reward we call "Real Reward" that prizes the agent when reaching the goal while penalizing large actions;

(ii) $r_1(s, a) = -(a - \tilde{\pi}(s))^2$: a dense reward we call

[7] We assumed to have a sufficiently high number of samples in the forward learning phase to reach the asymptotic behavior $\epsilon(M, \gamma) \to 0$. Furthermore parameters M and N are numerically interchangeable, and for simplicity in Figure 1 and 3, we set a given value to N and $M = \infty$.

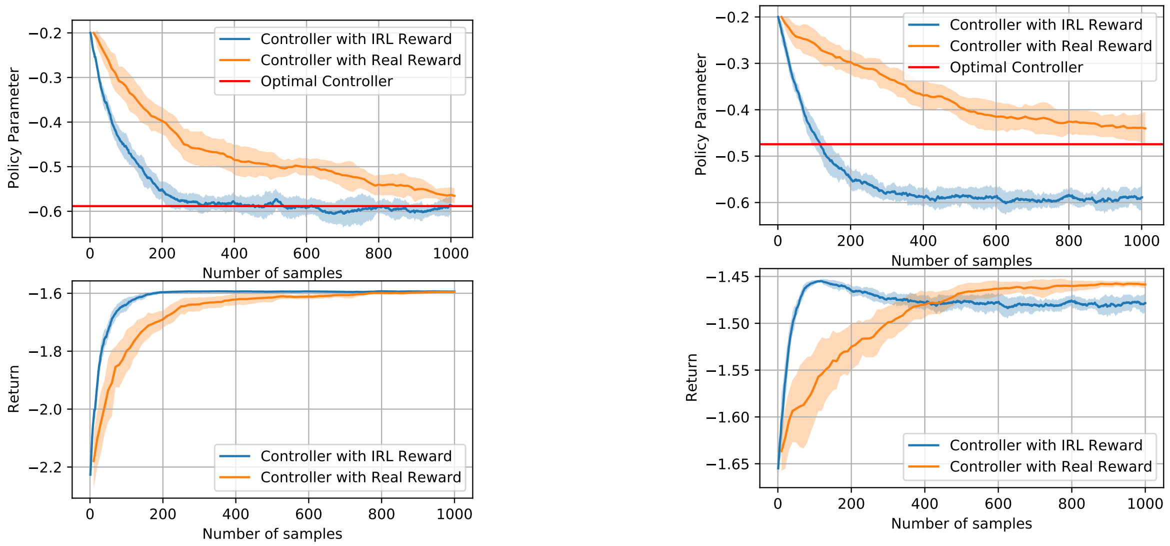


Figure 2: Comparison of the learned policy parameter and average return when learning in the same environment used for IRL (left) and when changing the environment (right) (10 runs, 95% c.i.).

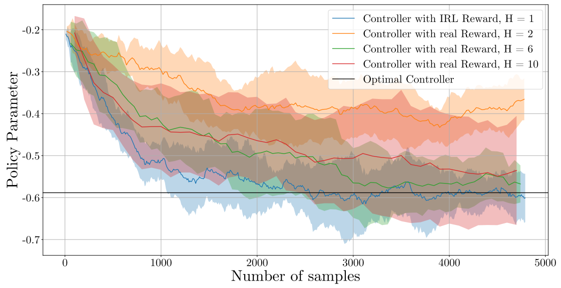


Figure 3: Impact of the optimized IRL reward on the sample efficiency of the forward learning task, and convergence to the expert's policy parameter (10 runs, 95% c.i.).

"IRL Reward", a *dense* reward that induces the agent to *imitate* a sub-optimal policy $\tilde{\pi}$, obtained by perturbing the parameters of the optimal policy.

While the "Real Reward", leading to the optimal policy, requires a large γ (because of the action penalization $-a^2$) and it is preferred by our IRL objective when many samples are available (i.e., large value for M), the "IRL Reward" leads to a sub-optimal policy but it is very easy to learn (it amounts to imitate $\tilde{\pi}$), admitting very small γ, and it is preferred for small values of M. Figure 4 shows the forward learning results obtained by REINFORCE on the two rewards and confirms the properties discussed above.

8. Conclusions

In this paper, we proposed a novel approach to the IRL problem which takes into account, during the reward selection phase, the availability of finite samples in the subsequent forward RL phase. The core idea was to select both the reward parameters and the discount factor through the definition of a min-max problem that minimizes the distance between the expert's policy and the learned policy in the successive forward learning task. In this way, the algorithm is able to find a trade-off between a potentially sub-optimal reward and the estimation error caused by using a finite number of samples in the forward learning phase. The numerical simulations showed the need for selecting a reward function accounting for the available samples in the successive forward RL phase and illustrated the features of our approach. Future works include the extension of the presented approach to more complex and challenging environments.

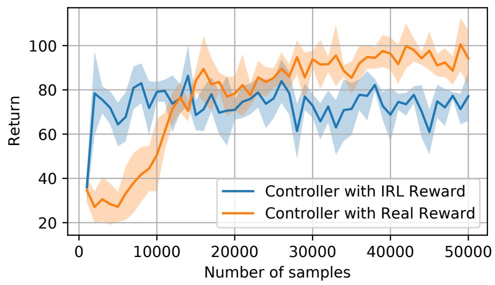


Figure 4: Comparison of the effect of two different reward functions on the sample complexity of REINFORCE on the Mountain Car problem (5 runs, 95% c.i.).

Acknowledgements

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

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. *Proceedings, Twenty-First International Conference on Machine Learning (ICML)*, pp. 1–8, 2004. doi: 10.1145/1015330.1015430.

Azar, M. G., Munos, R., and Kappen, H. J. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91(3): 325–349, 2013. doi: 10.1007/s10994-013-5368-1.

Bagnell, J. A. An invitation to imitation. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst, 2015.

Bain, M. and Sammut, C. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.

Boularias, A., Kober, J., and Peters, J. Relative entropy inverse reinforcement learning. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 182–189, 2011.

Bradtke, S. J. and Barto, A. G. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

Danskin, J. M. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966.

Dorato, P., Cerone, V., and Abdallah, C. *Linear-quadratic control: an introduction*. Simon & Schuster, Inc., 1994.

Farahmand, A. M., Munos, R., and Szepesvári, C. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 568–576, 2010.

Herman, M., Gindele, T., Wagner, J., Schmitt, F., and Burgard, W. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 51, pp. 102–110. PMLR, 2016.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. 4565–4573, 2016.

Ho, J., Gupta, J. K., and Ermon, S. Model-free imitation learning with policy optimization. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pp. 2760–2769, 2016.

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, UCL (University College London), 2003.

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., and Hebert, M. Activity forecasting. In *European Conference on Computer Vision*, pp. 201–214. Springer, 2012.

Krantz, S. G. and Parks, H. R. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2012.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4: 1107–1149, 2003.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13:3041–3074, 2012.

Likmeta, A., Metelli, A. M., Ramponi, G., Tirinzoni, A., Giuliani, M., and Restelli, M. Dealing with multiple experts and non-stationarity in inverse reinforcement learning: an application to real-life problems. *Mach. Learn.*, 110(9):2541–2576, 2021. doi: 10.1007/s10994-020-05939-8.

Lopes, M., Melo, F., and Montesano, L. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 31–46. Springer, 2009.

Metelli, A. M., Pirotta, M., and Restelli, M. Compatible reward inverse reinforcement learning. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pp. 2050–2059, 2017.

Metelli, A. M., Pirotta, M., and Restelli, M. On the use of the policy gradient and hessian in inverse reinforcement learning. *Intelligenza Artificiale*, 14(1):117–150, 2020. doi: 10.3233/IA-180011.

Moore, A. W. Efficient memory-based learning for robot control. Technical report, University of Cambridge, 1990.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.

Nedić, A. and Bertsekas, D. P. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13(1):79–110, 2003.

Neu, G. and Szepesvári, C. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 295–302, 2007.

Neu, G. and Szepesvári, C. Training parsers by inverse reinforcement learning. *Machine Learning*, 77(2-3):303, 2009.

Ng, A. Y. and Russell, S. J. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pp. 663–670. Morgan Kaufmann Publishers Inc., 2000.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., and Peters, J. An algorithmic perspective on imitation learning. *Found. Trends Robotics*, 7(1-2):1–179, 2018. doi: 10.1561/2300000053.

Pirotta, M. and Restelli, M. Inverse reinforcement learning through policy gradient minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 30, 2016.

Pirotta, M., Restelli, M., and Bascetta, L. Policy gradient in lipschitz markov decision processes. *Mach. Learn.*, 100 (2-3):255–283, 2015. doi: 10.1007/s10994-015-5484-1.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

Ramachandran, D. and Amir, E. Bayesian inverse reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2586–2591, 2007.

Ramponi, G., Likmeta, A., Metelli, A. M., Tirinzoni, A., and Restelli, M. Truly batch model-free inverse reinforcement learning about multiple intentions. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2359–2369, 2020.

Ratliff, N. D., Andrew Bagnell, J., and Zinkevic, M. A. Maximum margin planning. *ACM International Conference Proceeding Series*, 148:729–736, 2006. doi: 10.1145/1143844.1143936.

Ratliff, N. D., Silver, D., and Bagnell, J. A. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

Razaviyayn, M., Huang, T., Lu, S., Nouiehed, M., Sanjabi, M., and Hong, M. Non-convex min-max optimization: Applications, challenges, and recent theoretical advances. *arXiv:2006.08141*, Aug 2020. arXiv: 2006.08141.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, pp. 1889–1897, 2015.

Shiarlis, K., Messias, J., and Whiteson, S. Inverse reinforcement learning from failure. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pp. 1060–1068, 2016.

Silver, D., Bagnell, J. A., and Stentz, A. Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29(12):1565–1592, 2010.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12 (NIPS)*, pp. 1057–1063, 1999.

Syed, U., Bowling, M., and Schapire, R. E. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning (ICML)*, pp. 1032–1039. ACM Press, 2008a. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390286.

Syed, U., Bowling, M. H., and Schapire, R. E. Apprenticeship learning using linear programming. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, pp. 1032–1039, 2008b.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Wang, Y. and Li, J. Improved algorithms for convex-concave minimax optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. doi: 10.1007/BF00992696.

Ziebart, B. D. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, University of Washington, 2010.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum Entropy Inverse Reinforcement Learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, volume 8, pp. 1433—-1438. AAAI Press, 2008.

A. Proof of Proposition 4.3

The idea of this proposition is to start from the constraint (4):

$$\sum_{s\in\mathcal{S}} \widehat{Q}^{\pi_E}_M(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_M(s,\pi_E(s)) \geq 0, \tag{15}$$

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

$$\begin{aligned}
\widehat{Q}^{\pi_E}_M(s,\pi_{\boldsymbol{\eta}}(s)) &= \widehat{Q}^{\pi_E}_N(s,\pi_{\boldsymbol{\eta}}(s)) + \Big(\widehat{Q}\pi_E(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_N(s,\pi_{\boldsymbol{\eta}}(s))\Big) + \Big(\widehat{Q}^{\pi_E}_M(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))\Big) && (16)\\
&\leq \widehat{Q}^{\pi_E}_N(s,\pi_{\boldsymbol{\eta}}(s)) + \Big|\widehat{Q}\pi_E(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_N(s,\pi_{\boldsymbol{\eta}}(s))\Big| + \Big|\widehat{Q}^{\pi_E}_M(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}(s,\pi_{\boldsymbol{\eta}}(s))\Big| && (17)\\
&\leq \widehat{Q}^{\pi_E}_N(s,\pi_{\boldsymbol{\eta}}(s)) + \epsilon_2(N,\gamma) + \epsilon_1(M,\gamma), && (18)
\end{aligned}$$

where in the last step we applied the assumptions (5) and (6). Similarly, we can proceed with latter term, and derive:

$$\begin{aligned}
\widehat{Q}^{\pi_E}_M(s,\pi_E(s)) &= \widehat{Q}^{\pi_E}_N(s,\pi_E(s)) + \Big(\widehat{Q}\pi_E(s,\pi_E(s)) - \widehat{Q}^{\pi_E}_N(s,\pi_E(s))\Big) + \Big(\widehat{Q}^{\pi_E}_M(s,\pi_E(s)) - \widehat{Q}^{\pi_E}(s,\pi_E(s))\Big) && (19)\\
&\geq \widehat{Q}^{\pi_E}_N(s,\pi_E(s)) - \Big|\widehat{Q}\pi_E(s,\pi_E(s)) - \widehat{Q}^{\pi_E}_N(s,\pi_E(s))\Big| - \Big|\widehat{Q}^{\pi_E}_M(s,\pi_E(s)) - \widehat{Q}^{\pi_E}(s,\pi_E(s))\Big| && (20)\\
&\geq \widehat{Q}^{\pi_E}_N(s,\pi_E(s)) - \epsilon_2(N,\gamma) - \epsilon_1(M,\gamma), && (21)
\end{aligned}$$

where again in the last step we applied the assumptions (5) and (6). Putting back together the computed upper and lower bound we obtain:

$$\sum_{s\in\mathcal{S}} \widehat{Q}^{\pi_E}_M(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_M(s,\pi_E(s)) \leq \sum_{s\in\mathcal{S}} \widehat{Q}^{\pi_E}_N(s,\pi_{\boldsymbol{\eta}}(s)) - \widehat{Q}^{\pi_E}_N(s,\pi_E(s)) + 2\epsilon_2(N,\gamma) + 2\epsilon_1(M,\gamma). \tag{22}$$

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