Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Inverse Reinforcement Learning through Policy Gradient Minimization

Matteo Pirotta and **Marcello Restelli**
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Piazza Leonardo da Vinci, 32
I-20133, Milan, Italy
{matteo.pirotta, marcello.restelli}@polimi.it

Abstract

Inverse Reinforcement Learning (IRL) deals with the problem of recovering the reward function optimized by an expert given a set of demonstrations of the expert's policy. Most IRL algorithms need to repeatedly compute the optimal policy for different reward functions. This paper proposes a new IRL approach that allows to recover the reward function without the need of solving any "direct" RL problem. The idea is to find the reward function that minimizes the gradient of a parameterized representation of the expert's policy. In particular, when the reward function can be represented as a linear combination of some basis functions, we will show that the aforementioned optimization problem can be efficiently solved. We present an empirical evaluation of the proposed approach on a multidimensional version of the Linear-Quadratic Regulator (LQR) both in the case where the parameters of the expert's policy are known and in the (more realistic) case where the parameters of the expert's policy need to be inferred from the expert's demonstrations. Finally, the algorithm is compared against the state-of-the-art on the mountain car domain, where the expert's policy is unknown.

Introduction

Markov Decision Processes (MDPs) are an effective mathematical tool in modeling decision making in uncertain dynamic environments, where tasks are simply defined by providing a reward function. However, in many real-world problems, even the specification of the reward function can be problematic and it is easier to provide demonstrations from a desired policy. Inverse Reinforcement Learning (IRL) aims at finding the reward function that is (implicitly or explicitly) optimized by the demonstrated policy.

The approach proposed in this paper is based on the following observation: given a differentiable parametric representation of the expert's policy, that is optimal w.r.t. to some unknown reward function $\mathcal{R}^E$, its policy gradient computed according to $\mathcal{R}^E$ is zero. So, given a parametrized representation of the reward function, we solve the IRL problem by searching the reward function that minimizes some norm of the policy gradient. We show how this result can be obtained using a model-free approach that is based only on the data provided by the expert's demonstrations. Contrary to most existing algorithms, the proposed approach does not require to repeatedly solve the "direct" MDP. Although no assumption on the reward model is needed, for the case of linear parametrization we provide an efficient algorithm for computing the solution to the optimization problem. Empirical results on the Linear-Quadratic Regulator (LQR) test case allow to evaluate the effectiveness of the proposed method in recovering the parameters of the reward function optimized by the expert both when the expert's policy is known and when it has to be estimated from demonstrated trajectories. Comparisons with the state-of-the-art is performed on the well-known mountain-car domain.

Preliminaries

A Markov Decision process without reward MDP$\backslash\mathcal{R}$ is defined by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, D \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is a Markovian transition model where $\mathcal{P}(s'|s,a)$ defines the transition density between state s and s' under action a, $\gamma \in [0,1)$ is the discount factor, and D is the distribution of the initial state. A stochastic policy is defined by a density distribution $\pi(\cdot;s)$ that specifies for each state s the density distribution over the action space $\mathcal{A}$.

We observe the behavior of an expert that follows a policy π^E that is optimal w.r.t. some reward function $\mathcal{R}^E$. We assume that $\mathcal{R}^E$ can be represented through a linear or non-linear function $\mathcal{R}(s,a;\boldsymbol{\omega})$, where $\boldsymbol{\omega} \in \mathbb{R}^q$.

We consider infinite horizon problems where the future rewards are exponentially discounted with γ. Policies can be ranked by their expected discounted reward starting from the state distribution D:

$$J_D(\pi) = \int_{\mathcal{S}} d_\mu^\pi(s) \int_{\mathcal{A}} \pi(a;s)\mathcal{R}(s,a;\boldsymbol{\omega})\mathrm{d}a\mathrm{d}s,$$

where d_μ^π is the γ–discounted feature state distribution for starting state distribution D (Sutton et al. 1999). In this work we limit our attention to parametrized *differentiable* policies $\pi(a;s,\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^d$. Where possible we will use the compact notation $\pi_{\boldsymbol{\theta}}$.

Gradient Inverse Reinforcement Learning

In the first scenario we consider the problem of recovering the reward function $\mathcal{R}^E$ when the expert's policy π^E

is *known*. Having access to the analytic formulation of the policy, we can use gradient information to derive a new IRL algorithm that does not need to solve the forward model. It is worth to stress that even knowing the expert's policy, behavioral cloning may not be the best solution to IRL problems. In particular, the reward is a more succinct and powerful information than the policy itself. The reward function can be used to generalize the expert's behavior to state space regions not covered by the samples or to new situations. For instance, the transition model can change and, as a consequence, the optimal policy may change as well. Furthermore, behavioral cloning cannot be applied whenever the expert demonstrates actions that cannot be executed by the learner (think at a humanoid robot that learns by observing a human expert). In all these cases, knowing the reward function, the optimal policy can be derived.

Standard policy gradient approaches require the policy to be stochastic in order to get rid of the knowledge of the transition model (Sutton et al. 1999). When the expert's policy is deterministic the model must be available or the policy must be forced to be stochastic[1]. For continuous state-action domains the latter approach can be easily implemented by adding zero-mean Gaussian noise. Instead the Gibbs model is suited for discrete actions because the stochasticity can be regulated by varying the temperature parameter.

Given a parametric (linear or non linear) reward function $\mathcal{R}(s,a;\boldsymbol{\omega})$, we can compute the associate policy gradient

$$\nabla_{\boldsymbol{\theta}} J\left(\pi_{\boldsymbol{\theta}}^E, \boldsymbol{\omega}\right) = \int_{\mathcal{S}} d_\mu^{\pi_{\boldsymbol{\theta}}^E}(s) \int_{\mathcal{A}} \nabla_{\boldsymbol{\theta}} \pi^E(a;s,\boldsymbol{\theta}) Q^{\pi^E}(s,a;\boldsymbol{\omega})\, \mathrm{d}a\mathrm{d}s.$$

We assume to have an analytic description of the expert's policy, but we have a limited set of demonstrations of the expert's behavior in the environment. Let denote by $\mathcal{D} = \{\tau_i\}_{i=1}^N$ the set of expert's trajectories. Then, the gradient can be estimated off-line using the N trajectories and any standard policy gradient algorithm: REINFORCE (Williams 1992), GPOMDP (Baxter and Bartlett 2001), natural gradient (Kakade 2002) or eNAC (Peters and Schaal 2008a).

If the policy performance $J(\pi,\boldsymbol{\omega})$ is *differentiable* w.r.t. the policy parameters $\boldsymbol{\theta}$ and the expert $\pi_{\boldsymbol{\theta}}^E$ is optimal w.r.t. a parametrization $\mathcal{R}(s,a,\boldsymbol{\omega}^E)$, the associated policy gradient is identically equal to zero. Clearly, the expert's policy $\pi_{\boldsymbol{\theta}}^E$ is a *stationary point* for $J\left(\pi,\boldsymbol{\omega}^E\right)$.

The Gradient IRL (GIRL) algorithm aims at finding a stationary point of $J\left(\pi_{\boldsymbol{\theta}}^E,\boldsymbol{\omega}\right)$ w.r.t. the reward parameter $\boldsymbol{\omega}$, that is, for any $x, y \geq 1$

$$\boldsymbol{\omega}^A = \arg\min_{\boldsymbol{\omega}} \mathcal{C}_x^y\left(\pi_{\boldsymbol{\theta}}^E,\boldsymbol{\omega}\right) = \arg\min_{\boldsymbol{\omega}} \frac{1}{y}\left\|\nabla_{\boldsymbol{\theta}} J\left(\pi_{\boldsymbol{\theta}}^E,\boldsymbol{\omega}\right)\right\|_x^y. \quad (1)$$

GIRL properties One of the key properties of the GIRL algorithm is the *convexity* of the objective function whenever the parametric reward model is convex w.r.t. $\boldsymbol{\omega}$.

[1]Recently a deterministic version of the policy gradient theorem has been provided in (Silver et al. 2014). However, it cannot be directly applied in this framework because it requires a stochastic policy for exploration.

Lemma 1 (Convexity of $\mathcal{C}_x^y$)**.** *Given a* convex *representation of the reward function $\mathcal{R}(s,a;\boldsymbol{\omega})$ w.r.t. the reward parameters, the objective function $\mathcal{C}_x^y$, with $x, y \geq 1$, is convex.*

This means that the optimization problem can be solved using any standard convex optimization approach. In this work we consider the case of squared ℓ^2–norm ($x = 2$ and $y = 2$) and a constrained gradient descent approach.

We want to stress the fact that the only requirement for the convexity of the optimization process is the convexity of the parametric class of rewards (that is free to be designed). We think that this class is big enough to be used in several real problems. Note, no assumption is posed on the real (expert's) reward. In this scenario (convex optimization function) there are no problems related to stationary points.

Meaning of GIRL As mentioned before, when the expert is optimal w.r.t. the parametric reward, the minimum of $\mathcal{C}_x^y\left(\pi_{\boldsymbol{\theta}}^E,\boldsymbol{\omega}\right)$ is attained by the expert's weights $\boldsymbol{\omega}^E$. On the other hand, there are several reasons for which the expert may be not optimal: I) the expert is optimizing a reward function that cannot be represented by the chosen reward parametrization; II) the expert exhibits a suboptimal policy; III) the samples available to estimate the gradient are not sufficient to obtain a reliable estimate. When the expert's policy is not optimal for any reward function in the chosen reward class, the solution $\boldsymbol{\omega}^A$ represents the minimum norm gradient, i.e., the reward that induces the minimum change in the policy parameters. In other words, GIRL tries to provide the reward weights in the chosen space that better explain the behavior of the expert's policy. Note that the result $\boldsymbol{\omega}^A$ is reliable as long as the norm of the gradient is small enough, otherwise the optimal policy associated to the given reward weights can be arbitrarily different than expert's policy.[2]

Linear Reward Settings

In this section we reformulate the GIRL algorithm in the case of linear parametrization of the reward function $\mathcal{R}(s,a;\boldsymbol{\omega})$. This interpretation comes from a multi-objective optimization (MOO) view of the IRL problem. When the reward is linearly parametrized we have that

$$J(\pi,\boldsymbol{\omega}) = \sum_{i=1}^q \boldsymbol{\omega}_i \mathop{\mathbb{E}}_{\substack{s_0 \sim D \\ a_t \sim \pi\ s_t \sim \mathcal{P}}} \left[\sum_{t=0}^\infty \gamma^t \phi_i(s_t,a_t)\right] = \sum_{i=1}^q \boldsymbol{\omega}_i \boldsymbol{J}_i(\pi) \quad (2)$$

where $\boldsymbol{J}(\pi) \in \mathbb{R}^q$ are the unconditional feature expectations under policy π and basis functions $\boldsymbol{\Phi}(s,a) \in \mathbb{R}^q$.

MOO Perspective Equation (2) can be interpreted as a *weighted sum* of the objective vector $\boldsymbol{J}(\pi)$. This view connects our approach to the search of the reward weights that make the expert Pareto optimal. The test for locally Pareto optimal solutions is an important aspect of many MOO algorithms and has been extemv nsively studied (Fliege and

[2]The threshold between reliable and unreliable results is problem dependent.

Svaiter 2000). It can be formulated as (Brown and Smith 2005), for any non–zero (convex) vector $\boldsymbol{\alpha} \succeq 0$

$$\boldsymbol{\alpha} \in \texttt{Null}\left(D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi\right)\right), \tag{3}$$

where $\texttt{Null}$ is the (right) null space of the Jacobian $D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi\right)$. The same idea was used by (Désidéri 2012) to define the concept of Pareto–stationary solution. This definition derives from the geometric interpretation of Equation (3), that is, a point is Pareto–stationary if there exists a convex combination of the individual gradients that generates an identically zero vector:

$$D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi\right)\boldsymbol{\alpha} = \mathbf{0}, \quad \sum_{i=1}^{q}\alpha_i = 1, \quad \alpha_i \geq 0 \;\; \forall i \in \{1,\ldots,q\}.$$

A solution that is Pareto optimal is also *Pareto–stationary* (Désidéri 2012). A simple and intuitive interpretation of the Pareto-stationary condition can be provided in the case of two objectives. When the solution lies outside the frontier it is possible to identify a set of directions (ascent cone) that simultaneously increase all the objectives. When the solution belongs to the Pareto frontier (i.e., $\exists(convex)\boldsymbol{\alpha} : D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi\right)\boldsymbol{\alpha} = \mathbf{0}$), the ascent cone is empty because any change in the parameters will cause the decrease of at least one objective. Geometrically, the gradient vectors related to the different reward features result coplanar and with contrasting directions.

From MOO to IRL Under generic nonlinear reward parametrizations the problem faced by GIRL is a simple unconstrained problem. In the case of linear reward parametrization the GIRL problem is severely ill-posed (invariant to scalar factors) and admits a trivial zero-solution. A common solution to overcome this ambiguity problem (Ng and Russell 2000; Syed and Schapire 2007) is to restrict (without loss of generality) the expert weights $\boldsymbol{\omega}^E$ to belong to the unit $(q-1)$-simplex $\Delta^{q-1} = \{\boldsymbol{\omega} \in \mathbb{R}^q : \|\boldsymbol{\omega}\|_1 = 1 \wedge \boldsymbol{\omega} \succeq 0\}$[3].

As a consequence, the convex combination $\boldsymbol{\alpha}$ represents the scalarization, i.e., the reward parameters. In particular, notice that $\boldsymbol{\alpha} \equiv \boldsymbol{\omega}$ coincides with the solution computed by GIRL. This result follows easily by noticing that

$$\begin{aligned}\left\|\nabla_{\boldsymbol{\theta}} J\left(\pi^E_{\boldsymbol{\theta}}, \boldsymbol{\omega}\right)\right\|_x &= \left\|\int_{\mathcal{S}} d_{\mu}^{\pi^E_{\boldsymbol{\theta}}}(s)\int_{\mathcal{A}}\nabla_{\boldsymbol{\theta}}\pi^E(a;s,\boldsymbol{\theta})\boldsymbol{J}^{\pi}(s,a)\mathrm{d}a\mathrm{d}s\cdot\boldsymbol{\omega}\right\|_x \\ &= \left\|D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right)\boldsymbol{\omega}\right\|_x,\end{aligned}$$

where $\boldsymbol{J}^{\pi}(s,a) \in \mathbb{R}^q$ is the vector of conditional feature expectations given $s_0 = s$ and $a_0 = a$. When the expert is optimal, the latter equation is made identically equal to zero by any vector that is in the (right) null space of $D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right)$. Otherwise, as mentioned in the previous section, the vector $\boldsymbol{\omega}$ represents the reward function that induces the minimum change in the policy.

[3]The symbol $\succeq$ denotes the component-wise inequality.

Geometric Interpretation We have seen the analytical interpretation of the reward weights. Now, if the expert is optimal w.r.t. some linear combination of the objectives, then she lies on the Pareto frontier. Geometrically, the reward weights are orthogonal to the hyperplane tangent to the Pareto frontier (identified by the individual gradients $\nabla_{\boldsymbol{\theta}}\boldsymbol{J}_i\left(\pi^E\right), i = 1,\ldots,q$). By exploiting the local information given by the individual gradients w.r.t. the expert parametrization $D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right)$ we can compute the tangent hyperplane and the associated scalarization weights. Such hyperplane can be identified by the q points associated to the Gram matrix of $D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right) = \left[\nabla_{\boldsymbol{\theta}}\boldsymbol{J}_1\left(\pi^E\right),\ldots,\nabla_{\boldsymbol{\theta}}\boldsymbol{J}_q\left(\pi^E\right)\right]$:

$$\boldsymbol{G} = \left(D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right)\right)^{\mathrm{T}} D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right).$$

If $\boldsymbol{G}$ is full rank, the q points univocally identify the tangent hyperplane. Since the expert's weights are orthogonal to such hyperplane, they are obtained by computing the null space of the matrix $\boldsymbol{G}$: $\boldsymbol{\omega} \in \texttt{Null}(\boldsymbol{G})$. Given the individual gradients, the complexity of obtaining the weights is $\mathcal{O}(q^2d + q^3)$. In the following, we will call this version of GIRL as PGIRL (Plane GIRL).

An Analysis of Linear Reward Settings

We initially state our analysis in the case of complete knowledge where we can compute every single term in exact form. Later, we will deal with approximations.

Handling Degenerancy We have already mentioned that the GIRL problem is ill-posed under linear reward parametrizations. Possible sources of degenerancy of Problem (1) are, for instance, constant reward functions, duplicated features or useless features. These problems are well-known design issues that have been already solved in literature (Neu and Szepesvári 2007). The batch nature of the GIRL problem allows to incorporate a phase of feature preprocessing. In that phase we eliminate linearly dependent features—including also the features that are never active under the given policy—and constant features.

In PGIRL, this phase is carried out on the gradient matrix. The rank of the Jacobian matrix $D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right)$ plays a fundamental role in the reconstruction of the reward weights. Recall that the rank is limited by the minimum matrix dimension. Since we are interested in the influence of the objectives on the rank, we consider the policy parameters to be linearly independent, or more precisely, the rows of the Jacobian matrix to be linearly independent. Moreover, as long as the number d of policy parameters is greater or equal than the number q of reward parameters ($q \leq d$), any deficiency in the rank is due to linear dependence among the objectives. As a consequence, the Jacobian matrix can be reduced in order to contain only columns that are linearly independent (e.g., by means of echelon transformation (Nakos and Joyner 1998)). The removed columns (objectives) do not affect the reward parametrization, i.e., they are associated to zero weights. More critical is the scenario in which $q > d$ because it may be not possible to obtain a unique solution. Practically, the policy is able to influence only a subset of

the objectives spanned by the chosen representation. Preprocessing of the gradient or Gram matrix cannot be carried out since the rank is upper bounded by the policy parameters and not by the reward ones. However, this problem can be overcome by considering additional constraints or directly in the design phase since either q and d are usually under the control of the user.

When the Gram matrix $\boldsymbol{G}$ is not full rank we do not have a unique solution. In this case the null space operator returns an orthonormal basis $\boldsymbol{Z}$ such that $D_{\boldsymbol{\theta}}\boldsymbol{J}\left(\pi^E\right) \cdot \boldsymbol{Z}$ is a null matrix. However, infinite vectors are spanned by the orthonormal basis. The constraint on the ℓ^1–norm is not sufficient for the selection of the reward parameters, an additional criterion is required. Several approaches have been defined in literature. For example, in order to select solution vectors that consider only a subset of reward features, it is possible to design a sparsification approach (in the experiments $\ell^{\frac{1}{2}}$–norm has been used). A different approach is suggested by the field of reward shaping, where the goal is to find the reward function that maximizes an information criterion (Ng, Harada, and Russell 1999; Wiewiora, Cottrell, and Elkan 2003).

Approximate Case When the state transition model is unknown, the gradients are estimated through trajectories using some model-free approach (Peters and Schaal 2008a). The estimation errors of the gradients imply an error in the expert weights estimated by PGIRL. In the following theorem we provide an upper bound to the ℓ^2-norm of the difference between the expert weights $\boldsymbol{\omega}^E$ and the weights $\boldsymbol{\omega}^A$ estimated by PGIRL, given ℓ^2-norm upper bounds on the gradient estimation errors.[4]

Theorem 2. *Let denote with $\hat{\mathbf{g}}_i$ the estimated value of $\nabla_{\boldsymbol{\theta}}\boldsymbol{J}_i\left(\pi^E\right)$. Under the assumption that the reward function maximized by the expert is a linear combination of basis functions $\boldsymbol{\Phi}(s,a)$ with weights $\boldsymbol{\omega}^E$ and that, for each i, $\left\|\nabla_{\boldsymbol{\theta}}\boldsymbol{J}_i\left(\pi^E\right) - \hat{\mathbf{g}}_i\right\|_2 \leq \epsilon_i$:*

$$\left\|\boldsymbol{\omega}^E - \boldsymbol{\omega}^A\right\|_2 \leq \sqrt{2\left(1 - \sqrt{1 - \left(\frac{\bar{\epsilon}}{\bar{\rho}}\right)^2}\right)},$$

where $\bar{\epsilon} = \max_i \epsilon_i$ and $\bar{\rho}$ is the radius of the largest $(q-1)$-dimensional ball inscribed in the convex hull of points $\hat{\mathbf{g}}_i$.

Notice that the above bound is valid only when $\bar{\epsilon} < \bar{\rho}$, otherwise the upper bound is the maximum distance between two vectors belonging to the unit simplex, that is $\sqrt{2}$. As expected, when the gradients are known exactly (i.e., $\epsilon_i = 0$ for all i), the term $\bar{\epsilon}$ is zero and consequently the upper bound is zero. On the other hand, it is interesting to notice that a small value of the term $\bar{\epsilon}$ is not enough to guarantee that the weights computed by PGIRL are a good approximation of the expert ones. In fact, when $\bar{\rho} = \bar{\epsilon}$ the upper bound on the weight error reaches its maximum value. Intuitively, a small value of $\bar{\rho}$ means that all the gradients are aligned along some direction, thus making the problem less robust to approximation errors.

[4]The reader may refer to (Pirotta, Restelli, and Bascetta 2013) for error bounds on the gradient estimate with Gaussian policies.

Approximated Expert's Policy Parameters

When the expert's policy is unknown, to apply the same idea presented in the previous section, we need to infer a parametric policy model from a set of trajectories $\{\tau_i\}_{i=0}^N$ of length M. This problem is a standard density estimation problem. Given a parametric policy model $\pi(a; s, \boldsymbol{\theta})$, a parametric density estimation problem can be defined as a maximum likelihood estimation (MLE) problem: $\widehat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{N \cdot M} \pi(a_i; s_i, \boldsymbol{\theta})$.

Related Work

In the last decade, many IRL algorithms have been proposed (see (Zhifei and Meng Joo 2012) for a recent survey), most of which are approaches that require the MDP model and/or need to iteratively compute the optimal policy of the MDP obtained by considering intermediate reward functions. Since an accurate solution to the IRL problem may require many iterations, when the optimal policy for the MDP cannot be efficiently computed, these IRL approaches are not practical. For this reason, some recent works have proposed model-free IRL approaches that do not require to solve MDPs. The approach proposed by (Dvijotham and Todorov 2010) does not need to solve many MDPs, but it can be applied only to *linearly solvable* MDPs. In (Boularias, Kober, and Peters 2011) the authors proposed a model-free version of the Maximum Entropy IRL approach (Ziebart et al. 2008) that minimizes the *relative entropy* between the empirical distribution of the state-action trajectories demonstrated by the expert and their distribution under the learned policy. Even if this approach avoids the need of solving MDPs, it requires sampling trajectories according to a non-expert (explorative) policy. Classification-based approaches (SCIRL and CSI) (Klein et al. 2012; 2013) can produce near-optimal results when accurate estimation of the feature expectations can be computed and heuristic versions have been proved effective even when a few demonstrations are available. While SCIRL is limited to linearly parametrized reward functions, CSI can deal with nonlinear functions. However, both the algorithms require the expert to be deterministic and need to use heuristic approaches in order to learn with the only knowledge of expert's trajectories. Without using heuristics, they require an additional data set for exploration of the system dynamics. Moreover, CSI does not aim to recover the unknown expert's reward, but to obtain a reward for which the expert is nearly-optimal. Finally, the main *limitation* of these classification-based approaches is the assumption that the expert is optimal, the *suboptimality* scenario was *not* addressed.

In (Neu and Szepesvári 2007) the authors have proposed an IRL approach that can work even with non-linear reward parametrizations, but it needs to estimate the optimal action-value function. Other related works are (Levine and Koltun 2012) and (Johnson, Aghasadeghi, and Bretl 2013). Both approaches optimize an objective function related to the gradient information, but they require to know the dynamics.

Episodes	PGIRL					GIRL				
	R	RB	G	GB	eNAC	R	RB	G	GB	eNAC
10	0.031 ±0.011	0.020 ±0.0011	0.020 ±0.002	0.021 ±0.002	0.037 ±0.009	10.1 ±1.5	8.4 ±1.1	46.1 ±6.2	8.0 ±1.3	3.2 ±0.3
100	0.173 ±0.007	0.168 ±0.001	0.166 ±0.003	0.170 ±0.003	0.179 ±0.003	74.0 ±19.8	70.2 ±9.8	249.7 ±73.6	50.4 ±3.3	32.0 ±6.5
1000	1.664 ±0.006	1.655 ±0.011	1.655 ±0.031	1.683 ±0.005	1.640 ±0.038	971.4 ±270.0	472.7 ±25.2	3613.0 ±1128.2	498.6 ±32.5	257.4 ±14.7

Table 1: Average Computational Time (s) for the generation of the results presented in Figure 2b (5D LQR).

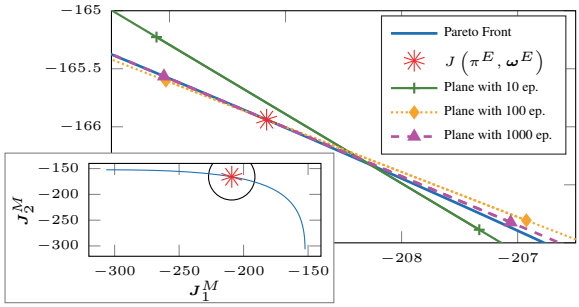


Figure 1: Behavior of the eNAC-PGIRL in 2D LQR. Figure reports the planes in the objective space identified by the PGIRL algorithm with 10, 100 and $1,000$ trajectories. This figure represents a zoom of the frontier around the current solution. The entire frontier is shown in the corner figure.

Experiments

This section is devoted to the empirical analysis of the proposed algorithms. The first domain, a linear quadratic regulator, is used to illustrate the main characteristics of the proposed approach, while the mountain car domain is used to compare it against the most related approaches.

Linear Quadratic Regulator

In this section we provide a set of experiments in the well-known Linear Quadratic Regulator (LQR) problem (Peters and Schaal 2008b). These experiments are meant to be a proof of concept of our algorithm behavior. We consider the multi-dimensional, multi-objective version of the problem provided in (Pirotta, Parisi, and Restelli 2015), the reader may refer to it for the settings. We consider a linear parametrization of the reward function:
$\mathcal{R}(s, a; \boldsymbol{\omega}) = -\sum_{i=1}^{q} \boldsymbol{\omega}_i (s^{\mathrm{T}} Q_i s + a^{\mathrm{T}} R_i a)$.

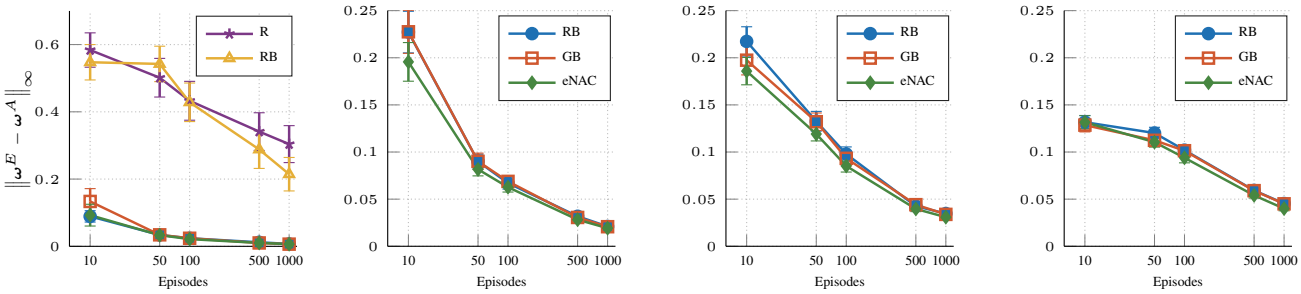
Exact Expert's Policy Parameters In the first test, we focus on the PGIRL approach where the gradient directions are computed using the eNAC algorithm (eNAC–PGIRL) in the 2D LQR domain. The goal is to provide a geometric interpretation of what the algorithm does. Figure 1 reports the planes (lines in 2D) and the associated weights obtained by the eNAC-PGIRL algorithm with different data set sizes. As the number of samples increases, the accuracy of the plane identified by the algorithm improves. With $1,000$ trajectories, the plane is almost tangent to the Pareto frontier. The points on the planes are obtained from the Gram matrix (after a translation from the origin).

The next set of experiments deals with the accuracy and time complexity of the proposed approaches (GIRL and PGIRL) with different gradient estimation methods (REINFORCE w/ and w/o baseline (RB, R), GPOMDP w/ and w/o baseline (GB,G) and eNAC). We selected 5 problem dimensions: (2, 5, 10, 20). For each domain we selected 20 random expert's weights in the unit simplex and we generated 5 different datasets. It is known that the contribute of the baseline for the gradient estimation is important and cannot be neglected (Peters and Schaal 2008b). Consider Figure 2a, using plain R and G the GIRL algorithm is not able to recover the correct weights. Although the error decreases as the number of trajectories increases, the error obtained with $1,000$ trajectories is larger than the one obtained by the baseline–versions (RB and GB) with only 10 trajectories. For this reason we have removed the plain gradient algorithms from the other tests. Figures 2b–2d replicate the test for increasing problem dimensions. All the algorithms show a decreasing error as the number of samples increases, but no significant differences can be observed. From such results, we can conclude that, when the expert's policy is known, GIRL is able to recover a good approximation of the reward function even with a few sample trajectories.

In Table 1 we show how the computational times of the different algorithms change as a function of the number of available trajectories. PGIRL algorithm outperforms GIRL for any possible configuration. Recall that PGIRL has to compute a fixed number of gradients, equal to the reward dimensionality, while GIRL is an iterative algorithm. We have imposed a maximum number of function evaluations to 500 for the convex optimization algorithm. The results show that the difference in the time complexity exceeds two orders of magnitude.[5] Although, the best choice for linear reward parametrizations is PGIRL, GIRL has the advantage of working even with non–linear parametrizations.

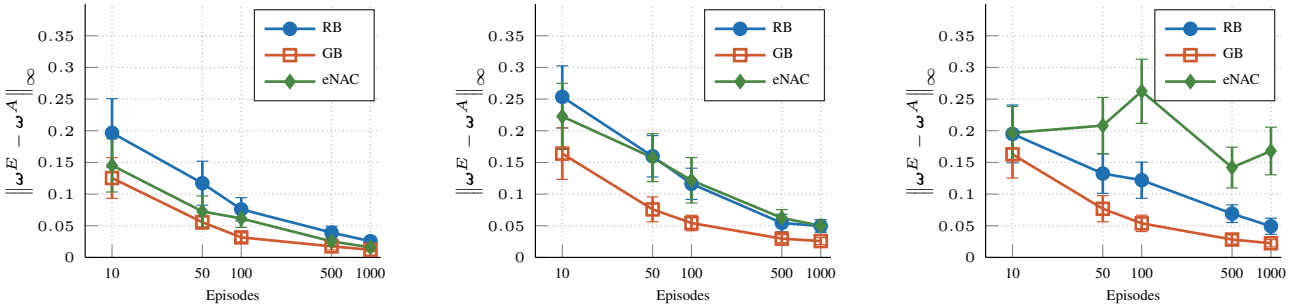
Approximated Expert's Policy Parameters In the following the parameters of the expert's policy are unknown and we have access only to expert's trajectories. In order to apply GIRL and PGIRL algorithms we have to learn a parametric policy from the data, that is, we have to solve a MLE problem (see Section). We consider a standard 1–dimensional LQR problem. Under these settings the policy is a Gaussian $a_t \sim \mathcal{N}(ks, \sigma^2)$ and the reward is $r_t = -\boldsymbol{\omega}_1 s_t^2 - \boldsymbol{\omega}_2 a_t^2$. The initial state is randomly selected in the interval $[-3, 3]$. Since the action space is continuous and

[5]The performance of the GIRL algorithm depends on the implementation of the convex algorithm and its parameters. Here we have exploited NLopt library (http://ab-initio.mit.edu/nlopt).



(a) 2D LQR (b) 5D LQR (c) 10D LQR (d) 20D LQR

Figure 2: The L_∞ norm between the expert's and agent's weights for different problem and data set dimensions. $s_0 = -\mathbf{10}$.



(a) Optimal Policy (linear) (b) Polynomial 3-deg (c) Gaussian RBFs

Figure 3: The L_∞ norm between the expert's and agent's weights are reported for different problem and data set dimensions.

we have to learn stochastic policies, we limit our research among the class of Gaussian policies with fixed diagonal covariance equal to $2 \cdot \boldsymbol{I}$. We consider three different mean parametrizations: linear in the state (i.e., the optimal one), with radial basis functions, and polynomial of degree 3.

Figure 3 reports the error made by the different policy parametrizations with different numbers of expert's trajectories. Even with a 3–degree polynomial approximation the learning process shown by the algorithm is smooth and the error decreases with the increase of the trajectory samples (Figure 3b). Finally, we consider the policy with 5 Gaussian RBFs uniformly placed in $[-4, 4]$ with an overlapping factor of 0.25. While REINFORCE and GPOMDP with baselines enjoy a smooth behavior, eNAC seems more affected by the estimation error in the policy parametrization and is not able to reduce the error even with $1,000$ trajectories.

Mountain Car

In order to compare the PGIRL algorithm against the state-of-the-art we consider the classical mountain car problem (Sutton et al. 1999). In particular we refer to the IRL version defined in (Klein et al. 2013) where REPS, CSI, SCIRL and a standard classifier have been compared. The same settings are used here except for the fact that the expert here is stochastic. She selects a random action with probability 0.1. It is worth to notice that this change has no impact on the other algorithms since REPS already uses a random dataset for importance sampling, while CSI and SCIRL exploit heuristics to provide samples for unseen actions. The expert's policy is provided in terms of trajectories, thus, an MLE estimate is required for PGIRL. We define the expert's policy as a Gibbs policy with linear approximation of the Q-function, a first degree polynomial over the state space is replicated for each action. The same features considered in (Klein et al. 2013) are used to represent the reward function over the state space: evenly-spaced hand-tuned 7×7 RBFs. Once the reward function is reconstructed, the new MDP is solved using LSPI (Lagoudakis and Parr 2003).

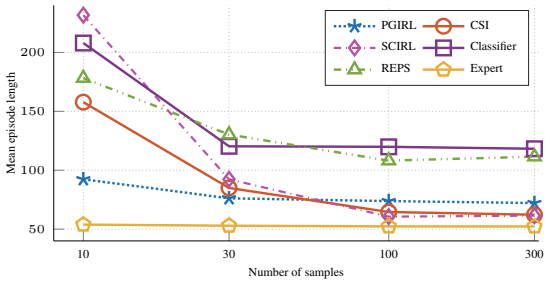


Figure 4: Mountain Car: algorithm performances.

The algorithms are evaluated based on the number of steps needed to reach the goal when starting from a random position in the valley, see Figure 4. We can see that PGIRL (using GPOMDP with baseline) is able to outperform the other algorithms when very few samples are available and attains

an intermediate value for increasing numbers of samples. By investigating the policies and the rewards recovered by PGIRL we have noticed that it favors lower velocities when approaching the goal. This explains why it takes slightly more steps than SCI and SCIRL. On the other side, CSI and SCIRL exploit more samples than PGIRL because, as mentioned before, they build additional samples using heuristics.

Conclusions and Future Work

We presented GIRL a novel inverse reinforcement learning approach that is able to recover the reward parameters by searching for the reward function that minimizes the policy gradient and avoids to solve the "direct" MDP. While GIRL defines a minimum–norm problem for linear and non-linear reward parametrizations, PGIRL method is an efficient implementation of GIRL in case of linear reward functions. We showed that, knowing the accuracy in the gradient estimates, it is possible to derive guarantees on the error between the recovered and the expert's reward weights. Finally, the algorithm has been evaluated on the LQR domain and compared with the state-of-the-art on the well-known mountain car domain. Tests show that GIRL represents a novel approach that efficiently and effectively solves IRL problems.

References

Baxter, J., and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research (JAIR)* 15:319–350.

Boularias, A.; Kober, J.; and Peters, J. 2011. Relative entropy inverse reinforcement learning. In *Proc. AISTATS*, 182–189.

Brown, M., and Smith, R. E. 2005. Directed multi-objective optimization. *International Journal of Computers, Systems, and Signals* 6(1):3–17.

Désidéri, J.-A. 2012. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique* 350(56):313 – 318.

Dvijotham, K., and Todorov, E. 2010. Inverse optimal control with linearly-solvable mdps. In *Proc. ICML*, 335–342.

Fliege, J., and Svaiter, B. F. 2000. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research* 51(3):479–494.

Johnson, M.; Aghasadeghi, N.; and Bretl, T. 2013. Inverse optimal control for deterministic continuous-time nonlinear systems. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, 2906–2913.

Kakade, S. M. 2002. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*. MIT Press. 1531–1538.

Klein, E.; Geist, M.; Piot, B.; and Pietquin, O. 2012. Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems*, 1007–1015.

Klein, E.; Piot, B.; Geist, M.; and Pietquin, O. 2013. A cascaded supervised learning approach to inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 1–16.

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

Levine, S., and Koltun, V. 2012. Continuous inverse optimal control with locally optimal examples. In *Proc. ICML*, 41–48.

Nakos, G., and Joyner, D. 1998. *Linear algebra with applications*. PWS Publishing Company.

Neu, G., and Szepesvári, C. 2007. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. UAI*, 295–302.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *Proc. ICML*, 663–670.

Ng, A. Y.; Harada, D.; and Russell, S. J. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proc. ICML*, 278–287.

Peters, J., and Schaal, S. 2008a. Natural actor-critic. *Neurocomputing* 71(79):1180 – 1190. Progress in Modeling, Theory, and Application of Computational Intelligenc 15th European Symposium on Artificial Neural Networks 2007 15th European Symposium on Artificial Neural Networks 2007.

Peters, J., and Schaal, S. 2008b. Reinforcement learning of motor skills with policy gradients. *Neural Networks* 21(4):682 – 697. Robotics and Neuroscience.

Pirotta, M.; Parisi, S.; and Restelli, M. 2015. Multiobjective reinforcement learning with continuous pareto frontier approximation. In *Proc. AAAI*, 2928–2934.

Pirotta, M.; Restelli, M.; and Bascetta, L. 2013. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems 26*, 1394–1402. Curran Associates, Inc.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. A. 2014. Deterministic policy gradient algorithms. In *Proc. ICML*, 387–395.

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

Syed, U., and Schapire, R. E. 2007. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems 20*, 1449–1456.

Wiewiora, E.; Cottrell, G. W.; and Elkan, C. 2003. Principled methods for advising reinforcement learning agents. In *Proc. ICML*, 792–799.

Williams, R. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8(3-4):229–256.

Zhifei, S., and Meng Joo, E. 2012. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics* 5(3):293–311.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proc. AAAI*, 1433–1438.