Provably Efficient Learning of Transferable Rewards

Alberto Maria Metelli [*1] **Giorgia Ramponi** [*1] **Alessandro Concetti** [1] **Marcello Restelli** [1]

Abstract

The reward function is widely accepted as a succinct, robust, and transferable representation of a task. Typical approaches, at the basis of Inverse Reinforcement Learning (IRL), leverage expert demonstrations to recover a reward function. In this paper, we study the theoretical properties of the class of reward functions that are compatible with the expert's behavior. We analyze how the limited knowledge of the expert's policy and of the environment affects the reward reconstruction phase. Then, we examine how the error propagates to the learned policy's performance when transferring the reward function to a different environment. We employ these findings to devise a provably efficient active sampling approach, aware of the need for transferring the reward function, that can be paired with a large variety of IRL algorithms. Finally, we provide numerical simulations on benchmark environments.

1. Introduction

Inverse Reinforcement Learning (IRL, Osa et al., 2018) aims at recovering a reward function by observing the behavior of an expert. One of the main challenges of IRL is that the problem itself is ill-posed, as multiple solutions are admissible (Ng & Russell, 2000). Several criteria have been proposed to address this *ambiguity* issue, based on different principles, including feature-based matching (Abbeel & Ng, 2004), maximum margin planning (Ratliff et al., 2006a), maximum entropy (Ziebart et al., 2008), maximum Hessian eigenvalue (Metelli et al., 2017), and generative adversarial learning (Ho & Ermon, 2016). These algorithms were evaluated either experimentally, in terms of performance of the policy learned using the recovered reward function (Ratliff et al., 2006a; Ziebart et al., 2008; Silver et al., 2010; Ziebart et al., 2010; Boularias et al., 2011; Ho & Ermon, 2016), or theoretically, under the strong assumption of reward uniqueness (Abbeel & Ng, 2004; Pirotta & Restelli, 2016; Ramponi et al., 2020b). Nevertheless, as noted in Osa et al. (2018), the evaluation of IRL algorithms remains, to a large extent, an open question.

Taking a step back, in the IRL framework, typically, the transition model of the underlying Markov Decision Process (MDP, Puterman, 2014) is unknown to the algorithm as well as the expert's policy. In general, these elements are estimated by interacting with the environment and by querying the expert. This leads to an unavoidable error on the *feasible set* of reward functions, i.e., the ones compatible with the expert's demonstrations. Motivated by this, the first question we aim to address is:

(Q1) *How does the error on the transition model and on the expert's policy propagate to the recovered reward?*

Clearly, any answer to this question will depend on the chosen IRL algorithm, i.e., on the criterion for selecting *one* reward function within the feasible set. To avoid the dependence on the specific IRL algorithm, we will address (Q1), studying directly the properties of the feasible set.

From an applicative point of view, the IRL's objective is twofold: *explainability* and *transferability*. On the one hand, understanding the expert's intentions is useful for descriptive purposes and can help interpret the expert's decisions (Russell & Santos, 2019; Juozapaitis et al., 2019; Hayat et al., 2019; Likmeta et al., 2021). On the other hand, the recovered reward function can be used to learn the same task in a possibly different environment (Abbeel & Ng, 2004; Levine et al., 2011; Fu et al., 2017). This ability makes the IRL approach more powerful than Behavioral Cloning (BC, Osa et al., 2018). Indeed, the reward function is the most "succinct" representation of a task (Sutton et al., 1998), and it can be *transferred* to other domains, unlike a *cloning* policy that is tightly connected to the environment in which it is played. These considerations motivate our second question:

(Q2) *How does the error on the recovered reward affect the performance of the policy learned in a different environment?*

Thus, the performance of the policy learned in the new environment will be our index for evaluating the quality

[*]Equal contribution [1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy. Correspondence to: Alberto Maria Metelli <albertomaria.metelli@polimi.it>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

of the recovered reward. As for question (Q1), we will not focus on a particular IRL algorithm but rather on the properties of the recovered feasible set of rewards.

Contributions The contributions of this paper can be summarized as follows.

- We study the error on the recovered reward by deriving a bound that highlights the individual contributions due to the estimation of the transition model and of the expert's policy (Section 3).
- We analyze the problem of transferring the reward to a new environment. We suppose to have a known *target* environment and to interact with a different *source* environment and with the corresponding expert's policy only. We derive a bound on the performance of the learned policy in the target environment when using the reward recovered from the source one (Section 4).
- Given the previous results, we consider a uniform sampling strategy for the IRL problem given a generative model of the source environment (Section 5). This leads to a sample complexity bound for IRL and makes a first step towards the answer of one of the open questions of this setting (Osa et al., 2018).
- Finally, taking inspiration from Zanette et al. (2019), we propose a new algorithm, *Transferable Reward ActiVE irL* (TRAVEL), that adapts the sampling strategy to the features of the problem. TRAVEL employs an IRL algorithm to choose the reward function from the estimated feasible set. We derive a problem-dependent upper bound to the sample complexity for the IRL setting (Section 6).

The proofs of the results presented in the main paper can be found in Appendix B.

2. Preliminaries

In this section, we introduce the background that will be employed throughout the remainder of the paper.

Mathematical Notation Let $\mathcal{X}$ be a finite set and $\mathcal{Y}$ be a space, we denote by $\mathcal{Y}^{\mathcal{X}}$ the set of functions $f:\mathcal{X}\to\mathcal{Y}$. The simplex over $\mathcal{X}$ is denoted by $\Delta^{\mathcal{X}}=\{\nu\in[0,1]^{\mathcal{X}}:\sum_{x\in\mathcal{X}}\nu(x)=1\}$ and we indicate with $\Delta^{\mathcal{Y}}_{\mathcal{X}}$ the set of functions $f:\mathcal{X}\to\Delta^{\mathcal{Y}}$. Let $\mu\in\Delta^{\mathcal{X}}$ and $f\in\mathbb{R}^{\mathcal{X}}$, we abbreviate with $\mu^\intercal f=\sum_{x\in\mathcal{X}}\mu(x)f(x)$, i.e., we use μ as an operator. We define the L_∞-norm of f as $\|f\|_\infty=\max_{x\in\mathcal{X}}|f(x)|$.

Markov Decision Processes A discounted Markov Decision Process without Reward function (MDP\R) is defined as a tuple $\mathcal{M}=(\mathcal{S},\mathcal{A},p,\gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p\in\Delta^{\mathcal{S}}_{\mathcal{S}\times\mathcal{A}}$ is the transition model, and $\gamma\in[0,1)$ is the discount factor. An MDP\R is a Markov Decision Process (MDP, Puterman, 2014) in which we remove the reward function. Given an MDP\R $\mathcal{M}$ and a reward function $r\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, we denote with $\mathcal{M}\cup r$ the MDP obtained by paring $\mathcal{M}$ and r. The agent's behavior is modeled by a policy $\pi\in\Delta^{\mathcal{A}}_{\mathcal{S}}$. We assume that the state and action spaces are finite with cardinality $S=|\mathcal{S}|$ and $A=|\mathcal{A}|$.

Operators Let $f\in\mathbb{R}^{\mathcal{S}}$ and $g\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$. We denote by P and π the operators induced by the transition model p and by the policy π, i.e., $(Pf)(s,a)=\sum_{s'\in\mathcal{S}}p(s'|s,a)f(s')$ and $(\pi g)(s)=\sum_{a\in\mathcal{A}}\pi(a|s)g(s,a)$. Moreover, we introduce the operator $(Ef)(s,a)=f(s)$. Given $\pi\in\Delta^{\mathcal{S}}_{\mathcal{S}\times\mathcal{A}}$, we denote with $(B^\pi g)(s,a)=g(s,a)\mathbb{1}\{\pi(a|s)>0\}$ and $(\overline{B}^\pi g)(s,a)=g(s,a)\mathbb{1}\{\pi(a|s)=0\}$. Finally, we denote the expectation under the discounted occupancy measure with $(I_{\mathcal{S}}-\gamma\pi P)^{-1}f=\sum_{t\in\mathbb{N}}(\gamma\pi P)^t f$. See Appendix A for a complete definition of the operators.

Value Functions and Optimality The *Q-function* $Q^\pi_{\mathcal{M}\cup r}\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ of a policy π in the MDP $\mathcal{M}\cup r$ is the expected discounted sum of the rewards starting from a state-action pair and playing policy π thereafter, and defined via the Bellman equation (Sutton et al., 1998): $Q^\pi_{\mathcal{M}\cup r}=r+\gamma P\pi Q^\pi_{\mathcal{M}\cup r}$. The *V-function* is the expectation of the Q-function over the action space: $V^\pi_{\mathcal{M}\cup r}=\pi Q^\pi_{\mathcal{M}\cup r}$. The *advantage function*, defined as $A^\pi_{\mathcal{M}\cup r}=Q^\pi_{\mathcal{M}\cup r}-EV^\pi_{\mathcal{M}\cup r}$, provides the one-step performance gain achieved by playing a specific action in a state rather than following policy π. A policy $\pi^*\in\Delta^{\mathcal{A}}_{\mathcal{S}}$ is optimal if it yields non-positive advantage, i.e., $A^{\pi^*}_{\mathcal{M}\cup r}(s,a)\leqslant 0$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$. We denote with $\Pi^*_{\mathcal{M}\cup r}\subseteq\Delta^{\mathcal{A}}_{\mathcal{S}}$ the set of optimal policies for the MDP $\mathcal{M}\cup r$. Under mild conditions (Puterman, 2014), any optimal policy attains the optimal V-function, i.e., $V^*_{\mathcal{M}\cup r}=\max_{\pi\in\Delta^{\mathcal{A}}_{\mathcal{S}}}V^\pi_{\mathcal{M}\cup r}$.

3. Recovering Feasible Rewards

In this section, we start by revising the formalization of the IRL problem. Then, we introduce and study the *feasible reward set*, i.e., the set of the reward functions that make the expert's policy optimal (Section 3.1). Finally, we analyze the error propagation when the feasible set is defined with an estimated transition model and expert's policy (Section 3.2). We start by formally stating the IRL problem using our notation and defining the feasible reward set.

Definition 3.1 (IRL Problem (Ng & Russell, 2000))**.** *An* Inverse Reinforcement Learning *(IRL) problem is a pair* $\mathfrak{P}=(\mathcal{M},\pi^E)$*, where* $\mathcal{M}$ *is an MDP\R and* $\pi^E\in\Delta^{\mathcal{A}}_{\mathcal{S}}$ *is an expert's policy. A reward* $r\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ *is* feasible *for* $\mathfrak{P}$ *if* π^E *is an optimal policy for the MDP* $\mathcal{M}\cup r$*, i.e.,* $\pi^E\in\Pi^*_{\mathcal{M}\cup r}$*. We denote by* $\mathcal{R}_{\mathfrak{P}}$ *the set of feasible rewards for* $\mathfrak{P}$*, named* feasible (reward) set.

As we pointed out in Section 1, the IRL problem admits multiple solutions, i.e., it suffers from an *ambiguity* issue. Thus, an IRL algorithm $\mathscr{A}$, implementing a criterion for

selecting a reward function within this set, can be seen as a *choice function* mapping an IRL problem $\mathfrak{P}$ to a feasible reward, i.e., $\mathscr{A}:\mathfrak{P}\mapsto r\in\mathcal{R}_{\mathfrak{P}}$. It is worth noting that the expert's reward function r^E (which is unknown) belongs to $\mathcal{R}_{\mathfrak{P}}$, but for the sake of the IRL problem, we are interested in finding just *one* reward inside $\mathcal{R}_{\mathfrak{P}}$. Thus, we primarily focus on the properties of the feasible set, rather than on specific criteria (i.e., IRL algorithms) for choosing a single reward within $\mathcal{R}_{\mathfrak{P}}$.

3.1. Feasible Reward Set

In this section, we study the properties of the feasible reward set. We start from the following result, which provides an *implicit* characterization of the feasible set.

Lemma 3.1 (Feasible Reward Set Implicit (Ng & Russell, 2000))**.** *Let $\mathfrak{P}=(\mathcal{M},\pi^E)$ be an IRL problem. Let $r\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, then r is a feasible reward, i.e., $r\in\mathcal{R}_{\mathfrak{P}}$ if and only if for all $(s,a)\in\mathcal{S}\times\mathcal{A}$ it holds that:*

$$\textit{(i)}\quad Q^{\pi^E}_{\mathcal{M}\cup r}(s,a)-V^{\pi^E}_{\mathcal{M}\cup r}(s)=0 \qquad \textit{if } \pi^E(a|s)>0,$$
$$\textit{(ii)}\quad Q^{\pi^E}_{\mathcal{M}\cup r}(s,a)-V^{\pi^E}_{\mathcal{M}\cup r}(s)\leqslant 0 \qquad \textit{if } \pi^E(a|s)=0.$$

Furthermore, if condition (ii) holds with the strict inequality, π^E is the unique optimal policy under r, i.e., $\Pi^*_{\mathcal{M}\cup r}=\{\pi^E\}$.

Both conditions are expressed in terms of the advantage function $A^{\pi^E}_{\mathcal{M}\cup r}(s,a)=Q^{\pi^E}_{\mathcal{M}\cup r}(s,a)-V^{\pi^E}_{\mathcal{M}\cup r}(s)$. Specifically, condition (i) prescribes that the advantage function of the actions that are played by the expert's policy π^E must be null, whereas condition (ii) ensures that the actions that are not played have a non-positive advantage. In other words, Lemma 3.1 requires π^E to be an optimal policy under reward function r. From Lemma 3.1, we derive an *explicit* form of the reward functions belonging to the feasible set.

Lemma 3.2 (Feasible Reward Set Explicit)**.** *Let $\mathfrak{P}=(\mathcal{M},\pi^E)$ be an IRL problem. Let $r\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, then r is a feasible reward, i.e., $r\in\mathcal{R}_{\mathfrak{P}}$ if and only if there exist $\zeta\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}_{\geqslant 0}$ and $V\in\mathbb{R}^{\mathcal{S}}$ such that:*

$$r=-\overline{B}^{\pi^E}\zeta+(E-\gamma P)V.$$

Thus, the reward function is the sum of two terms. The first term $-\overline{B}^{\pi^E}\zeta$ depends on the expert's policy π^E only but not on the MDP. It is zero for all actions the expert plays, i.e., those such that $\pi^E(a|s)>0$, while its value is non-positive for actions that the expert does not play, i.e., for those with $\pi^E(a|s)=0$. Requiring a strictly positive ζ allows enforcing π^E as the unique optimal policy. The second term $(E-\gamma P)V$ depends on the MDP but not on the expert's policy. It can be interpreted as a *reward-shaping* via function V, which is well-known to preserve the optimality of the expert's policy (Ng & Russell, 2000). By applying the Bellman equation, it is easy to see that the Q-function

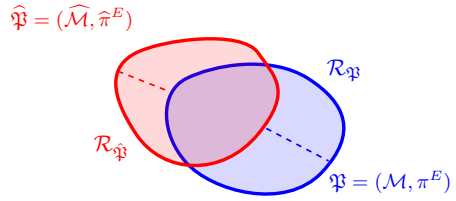

Figure 1. Feasible reward sets of two IRL problems: $\mathfrak{P}$ is an IRL problem and $\widehat{\mathfrak{P}}$ a version of $\mathfrak{P}$ estimated from samples.

induced by π^E under r is given by $Q^{\pi^E}_{\mathcal{M}\cup r}=-\overline{B}^{\pi^E}\zeta+EV$, in which V has the role of translating the Q-values by a fixed quantity within a the state. Thus, $-\overline{B}^{\pi^E}\zeta$ represents the advantage function $A^{\pi^E}_{\mathcal{M}\cup r}$ and V is the V-function $V^{\pi^E}_{\mathcal{M}\cup r}$.

3.2. Error Propagation in the Feasible Reward Set

We now study the *error propagation* in the feasible set, addressing question (Q1) of Section 1. Specifically, we consider two IRL problems $\mathfrak{P}=(\mathcal{M},\pi^E)$ and $\widehat{\mathfrak{P}}=(\widehat{\mathcal{M}},\widehat{\pi}^E)$. $\widehat{\mathfrak{P}}$ can be thought of as an approximate version of $\mathfrak{P}$, where the transition model $\widehat{p}$ and the expert's policy $\widehat{\pi}^E$ are estimated through samples. Intuitively, an error in estimating the transition model p and the expert's policy π^E results in an error in the estimation of the feasible sets $\mathcal{R}_{\mathfrak{P}}$. Since the IRL problem is ambiguous (it admits multiple solutions) we cannot expect to recover the expert's reward r^E exactly. Instead, we will be satisfied whenever we recover an accurate approximation of the feasible set, in which also r^E lies. Informally, we will say that the estimated feasible set $\mathcal{R}_{\widehat{\mathfrak{P}}}$ is "close" to the exact one $\mathcal{R}_{\mathfrak{P}}$ if for *every* reward $r\in\mathcal{R}_{\mathfrak{P}}$ there exists *one* estimated reward $\widehat{r}\in\mathcal{R}_{\widehat{\mathfrak{P}}}$ that is "close" to r and vice versa (Figure 1).[1] The following result formalizes the intuition by providing the error propagation result.

Theorem 3.1 (Error Propagation)**.** *Let $\mathfrak{P}=(\mathcal{M},\pi^E)$ and $\widehat{\mathfrak{P}}=(\widehat{\mathcal{M}},\widehat{\pi}^E)$ be two IRL problems. Then, for any $r\in\mathcal{R}_{\mathfrak{P}}$ such that $r=-\overline{B}^{\pi^E}\zeta+(E-\gamma P)V$ and $\|r\|_\infty\leqslant R_{\max}$ there exists $\widehat{r}\in\mathcal{R}_{\widehat{\mathfrak{P}}}$ such that element-wise it holds that:*

$$|r-\widehat{r}|\leqslant\overline{B}^{\pi^E}B^{\widehat{\pi}^E}\zeta+\gamma\left|\left(P-\widehat{P}\right)V\right|.$$

Furthermore, $\|\zeta\|_\infty\leqslant\frac{R_{\max}}{1-\gamma}$ and $\|V\|_\infty\leqslant\frac{R_{\max}}{1-\gamma}$.

The result states the *existence* of a reward $\widehat{r}$ in the estimated feasible set $\mathcal{R}_{\widehat{\mathfrak{P}}}$ fulfilling the bound that consists of two components. The first one $\overline{B}^{\pi^E}B^{\widehat{\pi}^E}\zeta$ depends on the policy approximation only. Specifically, this term is non-zero in the state-action pairs such that $\pi^E(a|s)=0$ and $\widehat{\pi}^E(a|s)>0$ only, i.e., for the actions that are not played by the expert but are wrongly believed to be played. Thus, to zero out this

[1] This notion of "closeness" between two sets is formalized by the *Hausdorff distance* (Rockafellar & Wets, 2009).

term it suffices to identify for each state *one* action played by the expert. The second term $|(P-\widehat{P})V|$, instead, concerns the estimation error of the transition model. Clearly, by reversing the roles of r and $\widehat{r}$, we can obtain a symmetric statement which, however, displays some differences when thinking to $\mathcal{R}_{\widehat{\mathfrak{P}}}$ as the estimated feasible set. Specifically, while the second term related to the transition model does not change (apart from V becoming $\widehat{V}$), the first one related to the policy becomes $\overline{B}^{\widehat{\pi}^E} B^{\pi^E}\widehat{\zeta}$. To zero out this term it is required identifying *all* actions played by the expert (not just one), that is a more demanding task. Clearly, this distinction vanishes for deterministic experts.

4. Transferring Rewards

As mentioned in Section 1, one of the advantages of IRL over BC is the possibility of reusing the learned reward function in a different environment. More specifically, we consider the following setting. There is an expert agent playing an optimal policy π^E in a *source* MDP\R $\mathcal{M}$. We want to recover a reward function explaining the expert's policy π^E in $\mathcal{M}$, knowing that we will employ it in a different *target* MDP\R $\mathcal{M}'$ for policy learning. In this section, we discuss the assumptions needed for transferring the reward function (Section 4.1). Then, we analyze how the error on the reward function propagates to the performance of the learned policy in the target environment (Section 4.2).

4.1. Transferable Reward Assumption

Transferring the recovered reward function poses new challenges that are quite unexplored in the IRL literature. Indeed, it might happen that different rewards are inducing the same expert's policy π^E in the source MDP\R $\mathcal{M}$, while generating different optimal policies in the target MDP\R $\mathcal{M}'$. More formally, let r^E be the true (and unknown) reward optimized by the expert's policy π^E and let $(\pi')^E$ the policy that the expert would play in $\mathcal{M}'$ optimizing the same r^E. Suppose we are able to solve the source IRL problem $\mathfrak{P}=(\mathcal{M},\pi^E)$ finding $r\in\mathcal{R}_{\mathfrak{P}}$, possibly different from r^E. There is no guarantee that r will make the policy $(\pi')^E$ optimal in the target MDP\R $\mathcal{M}'$. In other words, r might not be a solution to the target IRL problem $\mathfrak{P}'=(\mathcal{M}',(\pi')^E)$. In order to solve this additional ambiguity issue, we enforce the following assumption.

Assumption 4.1. *Let $\mathfrak{P}=(\mathcal{M},\pi^E)$ and $\mathfrak{P}'=(\mathcal{M}',(\pi')^E)$ be the source and target IRL problems. The corresponding feasible sets satisfy $\mathcal{R}_{\mathfrak{P}'}\supseteq\mathcal{R}_{\mathfrak{P}}$.*

With this assumption, we guarantee that every reward that is feasible for the source MDP\R $\mathcal{M}$ is also feasible for the target MDP\R $\mathcal{M}'$. We think that this assumption is unavoidable in our setting since we have no information regarding the *optimality* of the observed expert policy π^E in the target MDP\R $\mathcal{M}'$. The intuition behind Assumption 4.1 is that, by simply observing the expert playing an action in a state, we can only conclude that the action is optimal, but we are unable to judge "how much" suboptimal are the actions that the agent does not play. This issue could be overcome in two ways, which anyway require a modification of the setting and, thus, are out of the scope of this work: (i) we observe the optimal behavior of the agent in several different environments (Amin et al., 2017); (ii) we assume the expert plays a stochastic policy defined in terms of its Q-values. In this regard, Fu et al. (2017) proves that there is a one-to-one mapping between Boltzmann policies and Q-functions except for a state-only translation and scaling.

4.2. Error Propagation on the Value Function

In this section, we focus on question (Q2) presented in Section 1. We discuss, under Assumption 4.1, how an error on the reward function propagates into an error in estimating the optimal value function, and consequently on the optimal policy, when transferring the recovered reward to a possibly different MDP\R $\mathcal{M}'=(\mathcal{S},\mathcal{A},p',\gamma')$. We start with Lemma 4.1, which provides upper and lower bounds to the difference between the optimal Q-function under the true reward function $Q^*_{\mathcal{M}'\cup r}$ and the optimal Q-function under the estimated reward function $Q^*_{\mathcal{M}'\cup\widehat{r}}$.

Lemma 4.1 (Simulation Lemma 1)**.** *Let $\mathcal{M}'=(\mathcal{S},\mathcal{A},p',\gamma')$ be an MDP\R, let $r,\widehat{r}\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ be two reward functions. Then, for every $\pi^*\in\Pi^*_{\mathcal{M}'\cup r}$ and $\widehat{\pi}^*\in\Pi^*_{\mathcal{M}'\cup\widehat{r}}$ optimal policies for the MDPs $\mathcal{M}'\cup r$ and $\mathcal{M}'\cup\widehat{r}$ respectively, the following inequalities hold element-wise:*

$$Q^*_{\mathcal{M}'\cup r}-Q^*_{\mathcal{M}'\cup\widehat{r}}\leqslant\left(I_{\mathcal{S}\times\mathcal{A}}-\gamma' P'\pi^*\right)^{-1}(r-\widehat{r}),$$
$$Q^*_{\mathcal{M}'\cup r}-Q^*_{\mathcal{M}'\cup\widehat{r}}\geqslant\left(I_{\mathcal{S}\times\mathcal{A}}-\gamma' P'\widehat{\pi}^*\right)^{-1}(r-\widehat{r}).$$

In particular, it holds that:

$$\left\|Q^*_{\mathcal{M}'\cup r}-Q^*_{\mathcal{M}'\cup\widehat{r}}\right\|_\infty\leqslant\max_{\pi\in\{\pi^*,\widehat{\pi}^*\}}\left\|\left(I_{\mathcal{S}\times\mathcal{A}}-\gamma' P'\pi\right)^{-1}(r-\widehat{r})\right\|_\infty.$$

The result suggests that we need to be accurate in estimating the reward function of the state-action pairs that are highly visited by the discounted occupancy measures of the optimal policy $\pi^*\in\Pi^*_{\mathcal{M}'\cup r}$ and of the policy $\widehat{\pi}^*\in\Pi^*_{\mathcal{M}'\cup\widehat{r}}$ induced by the estimated reward function $\widehat{r}$. It is worth noting that Lemma 4.1 holds for *arbitrarily* chosen policies π^* and $\widehat{\pi}^*$ in the corresponding sets.

However, sometimes we are not interested in just estimating the value function accurately, rather in the performance of the policy $\widehat{\pi}^*$, computed with the estimated reward function $\widehat{r}$, under the true reward function r. The following result shows that we can reduce this problem to the one of estimating an accurate Q-function, as in Lemma 4.1.

Lemma 4.2 (Simulation Lemma 2)**.** *Let $\mathcal{M}'=(\mathcal{S},\mathcal{A},p',\gamma')$ be an MDP\R, let $r,\widehat{r}\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ be two reward functions, and*

let $\widehat{\pi}^* \in \Pi^*_{\mathcal{M}' \cup \widehat{r}}$ *be an optimal policy for* $\mathcal{M}' \cup \widehat{r}$*. Then, for every* $\pi^* \in \Pi^*_{\mathcal{M}' \cup r}$ *optimal policy for MDP* $\mathcal{M}' \cup r$*, the following inequality holds element-wise:*

$$V^*_{\mathcal{M}' \cup r} - V^{\widehat{\pi}^*}_{\mathcal{M}' \cup r} \leqslant \left(I_{\mathcal{S}} - \gamma' \widehat{\pi}^* P'\right)^{-1} (\pi^* - \widehat{\pi}^*) \times (Q^*_{\mathcal{M}' \cup r} - Q^*_{\mathcal{M}' \cup \widehat{r}}). \quad (1)$$

In particular, it holds that:

$$\left\| V^*_{\mathcal{M}' \cup r} - V^{\widehat{\pi}^*}_{\mathcal{M}' \cup r} \right\|_\infty \leqslant \frac{2}{1-\gamma'} \left\| Q^*_{\mathcal{M}' \cup r} - Q^*_{\mathcal{M}' \cup \widehat{r}} \right\|_\infty. \quad (2)$$

While the element-wise inequality in Equation (1) depends on the specific choice of $\widehat{\pi}^*$ (indeed while all $\widehat{\pi}^* \in \Pi^*_{\mathcal{M}' \cup \widehat{r}}$ induce the same value function under $\widehat{r}$ they might induce different value functions under r), the L_∞-norm inequality in Equation (2), although looser, does not depend on the specific policy $\widehat{\pi}^*$, but on the estimated reward $\widehat{r}$ only.

5. Learning Transferable Rewards with a Generative Model

In this section, we introduce the problem of learning a transferable reward function in a generative model setting and we introduce the notion of sampling strategy (Section 5.1). Then, we provide confidence intervals for the estimated transition model and expert's policy (Section 5.2). Finally, we study the sample complexity of a simple *uniform sampling* strategy (Section 5.3).

5.1. Problem Setting and Sampling Strategy

We start by explicitly stating the assumptions of the setting we consider.

Assumption 5.1. *The following statements hold:*

- *(i)* $\mathcal{M}$ *and* $\mathcal{M}'$ *have the same state and action spaces;*
- *(ii) we have access to the generative model of* $\mathcal{M}$*;*
- *(iii) we can query the expert's policy* π^E *in any state of* $\mathcal{M}$*;*
- *(iv) the expert's policy* π^E *is deterministic;*
- *(v) we know the transition model* p' *and the discount factor* γ' *of* $\mathcal{M}'$*.*

The sample collection proceeds at iterations. At each iteration $k \in [K]$, we collect at most $n_{\max} \in \mathbb{N}$ samples. When the generative model is queried about a state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, it responds with a transition triple (s,a,s'), where $s' \sim p(\cdot|s,a)$, and with an expert decision $\pi^E(s)$. The sampling strategy $\mathscr{S}$ decides, at each iteration k, how to allocate the $n_{\max}$ samples over the state-action space $\mathcal{S} \times \mathcal{A}$, with the goal of estimating the feasible set accurately. To this purpose, we introduce the following PAC requirement.

Definition 5.1. *Let* $\mathscr{S}$ *be a sampling strategy. Let* $\mathcal{R}_{\mathfrak{P}}$ *be the exact feasible set and* $\mathcal{R}_{\widehat{\mathfrak{P}}}$ *be the feasible set recovered after observing* $n \geqslant 0$ *samples collected in the source MDP*$\backslash$*R* $\mathcal{M}$*. Let* $(\overline{r}, \breve{r}) \in \mathcal{R}_{\mathfrak{P}} \times \mathcal{R}_{\widehat{\mathfrak{P}}}$ *be a pair of target rewards, we say that* $\mathscr{S}$ *is* (ϵ, δ, n)*-correct for MDP*$\backslash$*R* $\mathcal{M}'$ *and for the target rewards* $(\overline{r}, \breve{r})$ *if with probability at least* $1-\delta$ *it holds that:*

$$\inf_{\widehat{r} \in \mathcal{R}_{\widehat{\mathfrak{P}}}} \left\| Q^*_{\mathcal{M}' \cup \overline{r}} - Q^*_{\mathcal{M}' \cup \widehat{r}} \right\|_\infty \leqslant \epsilon$$
$$\inf_{r \in \mathcal{R}_{\mathfrak{P}}} \left\| Q^*_{\mathcal{M}' \cup \breve{r}} - Q^*_{\mathcal{M}' \cup r} \right\|_\infty \leqslant \epsilon. \quad (3)$$

The first condition guarantees that for a choice of the target reward in the exact feasible set $\overline{r} \in \mathcal{R}_{\mathfrak{P}}$ (for instance, the expert's one r^E), there exists a reward in the recovered feasible set $\widehat{r} \in \mathcal{R}_{\widehat{\mathfrak{P}}}$, inducing an ϵ-close Q-function. However, the first condition alone allows the presence of reward functions in $\mathcal{R}_{\widehat{\mathfrak{P}}}$ that do not explain any exact reward function (e.g., we could select $\mathcal{R}_{\widehat{\mathfrak{P}}} = \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$). Thus, we enforce the second condition, which guarantees that for a target reward function selected in the recovered feasible set $\breve{r} \in \mathcal{R}_{\widehat{\mathfrak{P}}}$ (for instance, the one produced by an IRL algorithm $\mathscr{A}(\mathcal{R}_{\widehat{\mathfrak{P}}})$), there always exists a reward function in the exact feasible set $r \in \mathcal{R}_{\widehat{\mathfrak{P}}}$ inducing an ϵ-close Q-function.[2]

The condition presented in Definition 5.1 refers to the ability to recover a reward function in the feasible set that induces an optimal Q-function $Q^*_{\mathcal{M}' \cup \widehat{r}}$ close to the one induced by a target reward $Q^*_{\mathcal{M}' \cup \overline{r}}$. As already mentioned in Section 4.2, we might be interested, instead, in evaluating the performance of an optimal policy $\widehat{\pi}^* \in \Pi^*_{\mathcal{M}' \cup \widehat{r}}$, recovered with the estimated reward function $\widehat{r}$, under a target reward function $\overline{r}$. In such a case, we have to focus on the value function difference $V^*_{\mathcal{M}' \cup \widehat{r}} - V^{\widehat{\pi}^*}_{\mathcal{M}' \cup \overline{r}}$. The following result, which makes use of Lemma 4.2, shows that fulfilling Definition 5.1, allows deriving guarantees for this case too.

Lemma 5.1. *Let* $\mathscr{S}$ *be a sampling strategy. Let* $\mathcal{R}_{\mathfrak{P}}$ *be the exact feasible set and* $\mathcal{R}_{\widehat{\mathfrak{P}}}$ *be the feasible set recovered after observing* $n \geqslant 0$ *samples collected in the source MDP*$\backslash$*R* $\mathcal{M}$*. Let* $(\overline{r}, \breve{r}) \in \mathcal{R}_{\mathfrak{P}} \times \mathcal{R}_{\widehat{\mathfrak{P}}}$ *be a pair of target rewards, if* $\mathscr{S}$ *is* (ϵ, δ, n)*-correct for MDP*$\backslash$*R* $\mathcal{M}'$ *and for the target rewards* $(\overline{r}, \breve{r})$*, as in Definition 5.1, then it holds that:*

$$\inf_{\widehat{r} \in \mathcal{R}_{\widehat{\mathfrak{P}}}} \sup_{\widehat{\pi}^* \in \Pi^*_{\mathcal{M}' \cup \widehat{r}}} \left\| V^*_{\mathcal{M}' \cup \overline{r}} - V^{\widehat{\pi}^*}_{\mathcal{M}' \cup \overline{r}} \right\|_\infty \leqslant \frac{2\epsilon}{1-\gamma'},$$
$$\inf_{r \in \mathcal{R}_{\mathfrak{P}}} \sup_{\pi^* \in \Pi^*_{\mathcal{M}' \cup r}} \left\| V^*_{\mathcal{M}' \cup \breve{r}} - V^{\pi^*}_{\mathcal{M}' \cup \breve{r}} \right\|_\infty \leqslant \frac{2\epsilon}{1-\gamma'}.$$

Thus, when moving from the approximation of the Q-functions $Q^*_{\mathcal{M}' \cup \overline{r}} - Q^*_{\mathcal{M}' \cup \widehat{r}}$, to the evaluation of performance of the learned policy $\widehat{\pi}^*$ under the target reward $\overline{r}$, we lose a factor $2/(1-\gamma')$. It is worth noting that the

[2] The *target* reward functions $(\overline{r}, \breve{r}) \in \mathcal{R}_{\mathfrak{P}} \times \mathcal{R}_{\widehat{\mathfrak{P}}}$ are a way of selecting *one* reward within the feasible set. As we shall see in the next sections, we will be able to provide sample-complexity guarantees for arbitrary choices of $(\overline{r}, \breve{r})$.

guarantee of Lemma 5.1 involves a supremum over the set of optimal policies for the candidate reward, i.e., $\Pi^*_{\mathcal{M}'\cup\widehat{r}}$, that helps discarding degenerate rewards, like constant ones.

5.2. Transition Model and Policy Estimation

For each iteration $k\in[K]$, we denote by $n_k(s,a,s')$ the number of times the triple $(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}$ is visited in episode k and $n_k(s,a)=\sum_{s'\in\mathcal{S}}n_k(s,a,s')$. For the transition model estimation, we define the cumulative counts $N_k(s,a,s')=\sum_{j\in[k]}n_j(s,a,s')$ and $N_k(s,a)=\sum_{j\in[k]}n_j(s,a)$, which lead to the estimate:

$$\widehat{p}_k(s'|s,a)=\frac{N_k(s,a,s')}{N_k^+(s,a)},$$

where $x^+=\max\{1,x\}$. Concerning the estimated expert's policy $\widehat{\pi}^E_k$, since the expert is deterministic for Assumption 5.1, the first time we sample a state $s\in\mathcal{S}$ we recover the true policy $\pi^E(s)$. We now show that, with high probability, we can guarantee an accurate estimate of the expert's policy and of the transition model.

Lemma 5.2 (Good Event)**.** *Let $\delta\in(0,1)$, define the* good event *$\mathcal{E}$ as the event such that the following inequalities hold simultaneously for all $(s,a)\in\mathcal{S}\times\mathcal{A}$ and $k\geqslant 1$:*

$$\left(\overline{B}^{\pi^E}B^{\widehat{\pi}^E_k}\zeta\right)(s,a)\leqslant\frac{R_{\max}}{1-\gamma}\mathbb{1}\{N_k(s)=0\},$$

$$\left(\overline{B}^{\widehat{\pi}^E_k}B^{\pi^E}\widehat{\zeta}_k\right)(s,a)\leqslant\frac{R_{\max}}{1-\gamma}\mathbb{1}\{N_k(s)=0\},$$

$$\left|\left(P-\widehat{P}_k\right)V\right|(s,a)\leqslant\frac{R_{\max}}{1-\gamma}\sqrt{\frac{2\ell_k(s,a)}{N_k^+(s,a)}},$$

$$\left|\left(P-\widehat{P}_k\right)\widehat{V}_k\right|(s,a)\leqslant\frac{R_{\max}}{1-\gamma}\sqrt{\frac{2\ell_k(s,a)}{N_k^+(s,a)}},$$

where ζ, $\widehat{\zeta}_k$, V, and $\widehat{V}_k$ are defined in Theorem 3.1 and $\ell_k(s,a)=\log\left(\frac{12SA(N_k^+(s,a))^2}{\delta}\right)$. Then, $\Pr(\mathcal{E})\geqslant 1-\delta$.

The terms related to the policy estimation become null as we identify the action played by the expert in each state, being the expert's policy deterministic. Therefore, they are replaced with $\mathbb{1}\{N_k(s)=0\}$. The terms related to the transition model, instead, are applications of Höeffding's inequality (Boucheron et al., 2013). By plugging the confidence interval of Lemma 5.2 into Theorem 3.1, we conclude that under the good event $\mathcal{E}$, at iteration $k+1$, given a target reward $\overline{r}\in\mathcal{R}_{\mathfrak{P}}$, there exists an estimated reward $\widehat{r}_{k+1}\in\mathcal{R}_{\widehat{\mathfrak{P}}_{k+1}}$ such that $|\overline{r}-\widehat{r}_{k+1}|(s,a)\leqslant\mathcal{C}_{k+1}(s,a)$ where:

$$\mathcal{C}_{k+1}(s,a)=\frac{R_{\max}}{1-\gamma}\left[\mathbb{1}\{N_{k+1}(s)=0\}+\gamma\sqrt{\frac{2\ell_{k+1}(s,a)}{N_{k+1}^+(s,a)}}\right],$$

and $N_{k+1}(s,a)=N_k(s,a)+n_{k+1}(s,a)$. Moreover, given a target reward $\breve{r}_{k+1}\in\mathcal{R}_{\widehat{\mathfrak{P}}_{k+1}}$, we can guarantee that under the same good event $\mathcal{E}$, there exists an exact reward function $r\in\mathcal{R}_{\mathfrak{P}}$ such that $|r-\breve{r}_{k+1}|(s,a)\leqslant\mathcal{C}_{k+1}(s,a)$ as well.

Algorithm 1 Uniform Sampling IRL

Input: significance $\delta\in(0,1)$, ϵ target accuracy, $n_{\max}$ maximum number of samples per iteration
$k\leftarrow 0$
$\epsilon_0=\frac{1}{1-\gamma'}$
while $\epsilon_k>\epsilon$ **do**
  Collect $\lceil\frac{n_{\max}}{SA}\rceil$ from each $(s,a)\in\mathcal{S}\times\mathcal{A}$
  Update accuracy $\epsilon_{k+1}=\frac{1}{1-\gamma'}\max_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{C}_{k+1}(s,a)$
  Update $\widehat{p}_{k+1}$ and $\widehat{\pi}^E_{k+1}$
  $k\leftarrow k+1$
end while

5.3. Uniform Sampling Strategy

The first algorithm we present employs a *uniform sampling* strategy to allocate samples over $\mathcal{S}\times\mathcal{A}$, until the desired accuracy $\epsilon>0$ is reached (Algorithm 1). The stopping condition makes use of the obtained confidence intervals:

$$\epsilon_{k+1}:=\frac{1}{1-\gamma'}\max_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathcal{C}_{k+1}(s,a)\leqslant\epsilon.$$

Thanks to Lemma 4.1, we are guaranteed that, under $\mathcal{E}$, when the stopping condition is activated, the recovered feasible set fulfills Definition 5.1, as shown below.

Theorem 5.1 (Sample Complexity of Uniform Sampling IRL)**.** *If Algorithm 1 stops at iteration K with accuracy ϵ_K, then with probability at least $1-\delta$ it fulfills Definition 5.1, for arbitrary target reward functions $\overline{r}$ and $\breve{r}$, with a number of samples upper bounded by:*

$$n\leqslant\widetilde{\mathcal{O}}\left(\frac{\gamma^2R_{\max}^2SA}{(1-\gamma')^2(1-\gamma)^2\epsilon_K^2}\right).$$

6. Active Learning of Transferable Rewards

In this section, we present a novel algorithm, named *Transferable Reward ActiVE irL* (TRAVEL), that adapts the sampling strategy to the structure of the problem. In order to choose which state-action pairs to sample from, we make use of Lemma 4.1. Suppose we are at iteration $k\in[K]$ and we have to decide how to allocate the $n_{\max}$ samples of the next iteration $k+1$. We have already observed that, under the good event $\mathcal{E}$, there exists $\widehat{r}_{k+1}\in\mathcal{R}_{\widehat{\mathfrak{P}}_{k+1}}$ and $r\in\mathcal{R}_{\mathfrak{P}}$ such that $|\overline{r}-\widehat{r}_{k+1}|(s,a)\leqslant\mathcal{C}_{k+1}(s,a)$ and $|\breve{r}_{k+1}-r|(s,a)\leqslant\mathcal{C}_{k+1}(s,a)$. Then, by Lemma 4.1 we have that:

$$\left\|Q^*_{\mathcal{M}'\cup\overline{r}}-Q^*_{\mathcal{M}'\cup\widehat{r}_{k+1}}\right\|_\infty\leqslant\max_{\pi\in\{\overline{\pi}^*,\widehat{\pi}^*_{k+1}\}}\left\|\left(I_{\mathcal{S}\times\mathcal{A}}-\gamma'P'\pi\right)^{-1}\mathcal{C}_{k+1}\right\|_\infty,$$

$$\left\|Q^*_{\mathcal{M}'\cup\breve{r}_{k+1}}-Q^*_{\mathcal{M}'\cup r}\right\|_\infty\leqslant\max_{\pi\in\{\breve{\pi}^*_{k+1},\pi^*\}}\left\|\left(I_{\mathcal{S}\times\mathcal{A}}-\gamma'P'\pi\right)^{-1}\mathcal{C}_{k+1}\right\|_\infty,$$

where the policies are *arbitrarily* selected in the corresponding sets: $\overline{\pi}^*\in\Pi^*_{\mathcal{M}'\cup\overline{r}}$, $\widehat{\pi}^*_{k+1}\in\Pi^*_{\mathcal{M}'\cup\widehat{r}_{k+1}}$, $\breve{\pi}^*_{k+1}\in\Pi^*_{\mathcal{M}'\cup\breve{r}_{k+1}}$, and $\pi^*\in\Pi^*_{\mathcal{M}'\cup r}$. In principle, we could optimize the right-

Algorithm 2 TRAVEL

Input: significance $\delta\in(0,1)$, ϵ target accuracy, $n_{\max}$ maximum number of samples per iteration, IRL algorithm $\mathscr{A}$
$k\leftarrow 0$
$\epsilon_0=\frac{1}{1-\gamma'}$
while $\epsilon_k>\epsilon$ **do**
  Solve optimization problem in Eq (4) for n_{k+1} and ϵ_{k+1}
  Collect $n_{k+1}(s,a)$ samples from $(s,a)\in\mathcal{S}\times\mathcal{A}$
  Update $\widehat{p}_{k+1}$ and $\widehat{\pi}^E_{k+1}$
  $k\leftarrow k+1$
end while

hand side of the previous inequalities over n_{k+1} to obtain the sample allocation. However, we have no knowledge about all the involved policies. Thus, we resort to a surrogate bound that leads to an allocation better than the uniform one. To this purpose, given an IRL algorithm $\mathscr{A}$, we follow the spirit of (Zanette et al., 2019) extending the maximization over a set of policies $\Pi^{\mathscr{A}}_k$ that, with high probability, contains the needed ones:

$$\Pi^{\mathscr{A}}_k=\left\{\pi\in\Delta^{\mathcal{A}}_{\mathcal{S}}\colon \sup_{\mu_0\in\Delta_{\mathcal{S}}}\mu_0{}^{\intercal}\left(V^*_{\mathcal{M}'\cup\mathscr{A}(\mathcal{R}_{\widehat{\mathfrak{P}}_k})}-V^{\pi}_{\mathcal{M}'\cup\mathscr{A}(\mathcal{R}_{\widehat{\mathfrak{P}}_k})}\right)\leqslant 4\epsilon_k\right\},$$

where the value of ϵ_k will be defined later. Here is the first point in which we actually make use of an IRL algorithm $\mathscr{A}$, whose goal is to choose a reward in the feasible reward set. The rationale in the definition of $\Pi^{\mathscr{A}}_k$ is to constrain the search for the policy to those yielding a value function at iteration k close to the estimated optimal one. We can now formulate the optimization problem:

$$\begin{aligned}\epsilon_{k+1}:=&\min_{n_{k+1}\in\mathbb{N}^{\mathcal{S}\times\mathcal{A}}}\max_{\substack{\mu_0\in\Delta^{\mathcal{S}\times\mathcal{A}}_{\mathcal{S}}\\ \pi\in\Delta^{\mathcal{A}}_{\mathcal{S}}}}\mu_0{}^{\intercal}\left(I_{\mathcal{S}\times\mathcal{A}}-\gamma' P'\pi\right)^{-1}\mathcal{C}_{k+1}\\ \text{s.t. }&\mu_0{}^{\intercal}E\left(V^*_{\mathcal{M}'\cup\mathscr{A}(\mathcal{R}_{\widehat{\mathfrak{P}}_k})}-V^{\pi}_{\mathcal{M}'\cup\mathscr{A}(\mathcal{R}_{\widehat{\mathfrak{P}}_k})}\right)\leqslant 4\epsilon_k\\ &\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}n_{k+1}(s,a)\leqslant n_{\max}.\end{aligned}\tag{4}$$

The program is a minimax in which we look for the sample allocation n_{k+1} that minimizes the bound on the value function difference of Lemma 4.1, under the worst possible policy π in the set $\Pi^{\mathscr{A}}_k$ and initial state-action distribution μ_0. Therefore ϵ_k, used to define $\Pi^{\mathscr{A}}_k$, is the objective function value of the previous iteration k. It can be proved that under the good event $\mathcal{E}$, $\Pi^{\mathscr{A}}_k$ contains a specimen of all the required optimal policies, i.e., $\overline{\pi}^*$, $\widehat{\pi}^*_{k+1}$, $\breve{\pi}^*_{k+1}$, and π^* (Corollary B.2). The constant $n_{\max}$ is the maximum number of samples allowed per iteration and it is a user-defined parameter. By choosing the $n_{\max}$ value, the user has to make a trade-off between time and sample efficiency. If the value of $n_{\max}$ is too high, the algorithm achieves the desired ϵ-correctness very quickly but with a possible sample inefficient behavior (close to uniform); if the value of $n_{\max}$ is too low, many iterations are needed to achieve the desired accuracy ϵ, but choosing more carefully where to sample. The pseudocode of TRAVEL is reported in Algorithm 2.

It is worth noting that we have not specified which IRL algorithm should be employed to recover a reward function. Indeed, any IRL algorithm $\mathscr{A}$ can be used for this purpose, provided that it selects a reward function within the feasible set $\mathcal{R}_{\widehat{\mathfrak{P}}}$. We stress that the main goal of this paper is not to provide a new IRL algorithm for choosing a good reward from the feasible reward set, but to explain how to recover a good approximation of this feasible set.

6.1. Sample Complexity

In this section, we prove that TRAVEL fulfills the PAC-condition of Definition 5.1. In order to provide this result, we use as suboptimality gaps the negative advantage: $-A^*_{\mathcal{M}'\cup\widetilde{r}}(s,a)=V^*_{\mathcal{M}'\cup\widetilde{r}}(s)-Q^*_{\mathcal{M}'\cup\widetilde{r}}(s,a)$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$, where $\widetilde{r}\in\operatorname{arginf}_{r\in\mathcal{R}_{\mathfrak{P}}}\|r-\mathscr{A}(\mathcal{R}_{\widehat{\mathfrak{P}}_K})\|_\infty$ is the reward function in the exact feasible set $\mathcal{R}_{\mathfrak{P}}$ closest to the one returned by the IRL algorithm $\mathscr{A}$ applied to the estimated feasible set $\mathcal{R}_{\widehat{\mathfrak{P}}_K}$.

Theorem 6.1 (Sample Complexity of TRAVEL)**.** *If Algorithm 2 stops at iteration K with accuracy ϵ_K and accuracy ϵ_{K-1} at the previous iteration, then with probability at least $1-\delta$ it fulfills Definition 5.1, for arbitrary target reward functions $\overline{r}$ and $\breve{r}$, with a number of samples upper bounded by $n=\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}N_K(s,a)$ where:*

$$N_K(s,a)\leqslant\widetilde{\mathcal{O}}\left(\min\left\{\frac{\gamma^2R^2_{\max}}{(1-\gamma')^2(1-\gamma)^2\epsilon^2_K},\frac{\gamma^2R^2_{\max}\epsilon^2_{K-1}}{(1-\gamma)^2(-A^*_{\mathcal{M}'\cup\widetilde{r}}(s,a))^2\epsilon^2_K}\right\}\right).$$

The significance of this result depends on two main components: the ratio between the two objectives ϵ_{K-1} and ϵ_K and the suboptimality gaps. The latter depends, although indirectly, on the employed IRL algorithm $\mathscr{A}$. The more suboptimal the action is, the less the action will be sampled. Instead, the $\epsilon_{K-1}/\epsilon_K$ component depends on the choice of the $n_{\max}$ value: if this value is small, then the ratio will also be small. As discussed in the previous section, if the $n_{\max}$ value is too high, the algorithm tends to sample every action-state pair uniformly.

6.2. Discussion

The upper bound on the sample complexity that we derive in the problem-independent version is of order $\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\gamma')^2(1-\gamma)^2\epsilon^2}\right)$. Although the problem we address is intimately different, it is interesting to compare this result with the well-known lower bound for RL with a generative model (Azar et al., 2013): $\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\gamma)^3\epsilon^2}\right)$, matched by several algorithms (Azar et al., 2012; Sidford et al., 2018; Zanette et al., 2019). Thus, when $\gamma'=\gamma$, we have an exponent 4 for the term $1/(1-\gamma)$. One might be tempted to think that this is a consequence of using Höeffding's inequality

instead of the tighter Bernstein's inequality.[3] We think that this might not be the case as the crucial property that allows achieving power 3 is a careful bound of the variance of $\widehat{P}V^*$ (Azar et al., 2013, Lemmas 7 and 8). There are two reasons why we cannot exploit this bound. First, we consider the expectations taken w.r.t. a different target model P', while the estimates are conducted on the source one P. Second, we are required, based on Theorem 3.1, to estimate the variance of $\widehat{P}V$ and V is unknown and hard to estimate.

7. Related Works

The IRL problem was introduced by Ng & Russell (2000). Most early IRL algorithms assume that the dynamics of the system are known. Many criteria were proposed for selecting a *good* reward function in the feasible reward set, based on features matching (Abbeel & Ng, 2004), maximum margin (Ratliff et al., 2006a), maximum entropy (Ziebart et al., 2008; 2010), Bayesian framework (Ramachandran & Amir, 2007), boosting methods (Ratliff et al., 2006b) and Gaussian processes (Levine et al., 2011). A limited number of IRL algorithms can be considered model-free: Relative Entropy Inverse Reinforcement Learning (Boularias et al., 2011), Generative Adversarial Imitation Learning (Ho & Ermon, 2016), Gradient-based Inverse Reinforcement Learning (Pirotta & Restelli, 2016) and its extensions (Metelli et al., 2017; 2020; Ramponi et al., 2020b;a). Other works on Imitation Learning use an active approach, such as the one used in this paper. Judah et al. (2012) draw a reduction from active imitation learning to i.i.d. active learning. In (Ross & Bagnell, 2010) and (Ross et al., 2011), the authors propose two approaches based on executing the estimated policy and asking an oracle for a dataset containing the action performed by the expert. In these papers, however, no guarantees on the sample complexity are provided. The closest work to ours is by Lopes et al. (2009), which propose a method to actively ask for samples from a generator to perform IRL, adopting a Bayesian approach. However, they assume knowledge of the real transition model and the main effort lies in estimating the expert's policy. Since we do not assume the knowledge of the transition model, this work is not fully comparable to our setting.

8. Experimental Evaluation

In this section, we provide the experimental evaluation of TRAVEL with a threefold goal. In the first experiment, we motivate the need for employing IRL over BC when our goal is to transfer knowledge to a target environment (Section 8.1). Then, we highlight the benefits of the sampling strategy of TRAVEL over Uniform Sampling (Section 8.2). Finally, we show how TRAVEL can be combined with different IRL algorithms (Section 8.3). In all the experiments, we employ reward functions that depend on the state only and the algorithms are evaluated according to the following performance index $\|V^*_{\mathcal{M}'\cup r^E} - V^{\widehat{\pi}^*}_{\mathcal{M}'\cup r^E}\|_2^2$, where the symbols are defined in the previous sections (we omit the subscript $\mathcal{M}'\cup$ in the plots). The complete experimental results are provided in Appendix D.

[3] In the RL with generative model setting, Höeffding's inequality leads to the same exponent 4.

8.1. IRL vs Behavioral Cloning

In this section, we show the benefits of IRL over BC when we want to learn a transferable reward function. In BC, the collected samples are used to estimate a policy describing the expert's behavior directly. The recovered policy is typically highly dependent on the environment in which the expert is acting, therefore, in many cases, it cannot be transferred to different environments. We use a 3×3 Gridworld environment with an obstacle in the central cell that makes the agent bouncing back with probability p and surpassing it with probability $1-p$. If $p\simeq0$ the optimal policy is to collide with the obstacle until the agent reaches the goal state. While, if $p\simeq1$, an optimal agent gets around the obstacle. The source MDP has obstacle's probability $p=0.8$ and target MDPs are four Gridworlds with obstacle's probabilities $p\in\{0,0.2,0.5,0.8\}$. For this experiment, we use a simple IRL algorithm that enforces the conditions of Lemma 3.1 and chooses the reward function to maximize the minimum action gap; we call it *MaxGap-IRL* (details in Appendix C). The results in Figure 2 show that the performance of BC deteriorates as the source and target MDPs become more dissimilar, as expected. Differently, TRAVEL combined with MaxGap-IRL allows recovering a reward function that leads to an optimal policy. Thus, as long as the target and source environments are the same (Figure 2 first plot) BC is a valid alternative, but IRL becomes unavoidable when the need for transferring knowledge arises.

8.2. TRAVEL vs Uniform Sampling

As discussed in Section 5.3, Uniform Sampling IRL and TRAVEL differ from the strategy used to allocate samples to the state-action pairs. While Uniform Sampling queries the generative model uniformly, TRAVEL actively allocates samples in the state-action pairs that will carry "more information". We consider a chain MDP composed by 6 states $\mathcal{S}=\{s_0,\dots,s_4,s_b\}$ and 10 actions $\mathcal{A}=\{a_g,a_1,\dots,a_9\}$ (Figure 3). We have tested both algorithms and the results are shown in Figure 4. Although Uniform Sampling IRL seems to perform better with a small number of samples, we observe that TRAVEL recovers a reward function that allows achieving a near-optimal performance in less than half of the samples needed by Uniform Sampling.

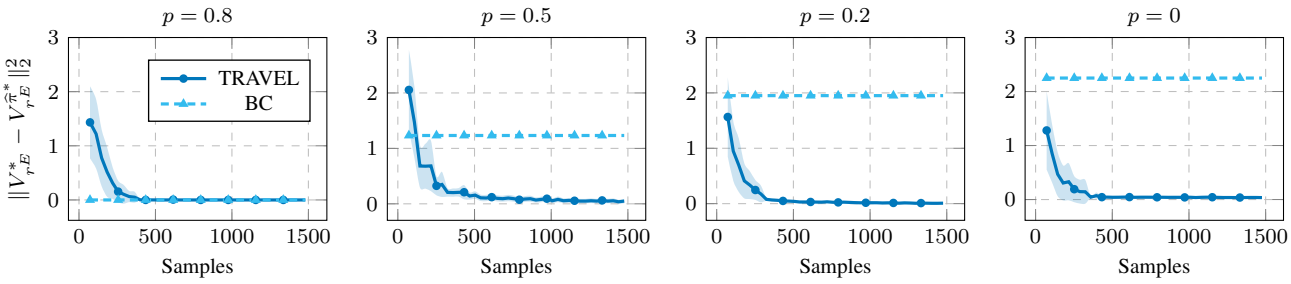

Figure 2. Comparison between TRAVEL and Behavioral Cloning (BC) on Gridworld environment, with different values of obstacle's probability for the target MDP $\mathcal{M}'$. 200 runs, 98% c.i.

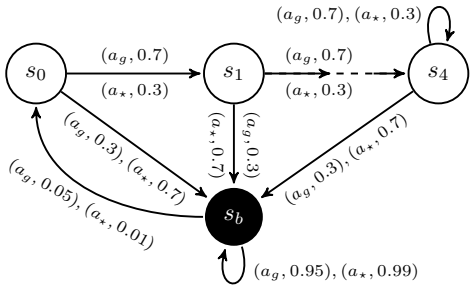

Figure 3. MDP employed in Section 8.2. States s_2 and s_3 behave exactly as s_1. $a_\star$ denotes any action in $\{a_1, \ldots, a_9\}$.

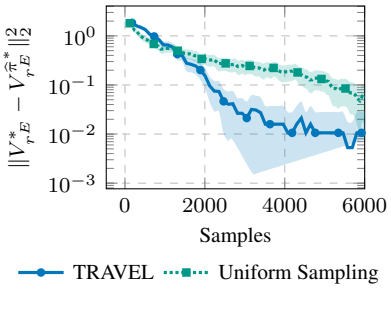

Figure 4. Comparison between Uniform Sampling IRL and TRAVEL. 300 runs, 98% c.i.

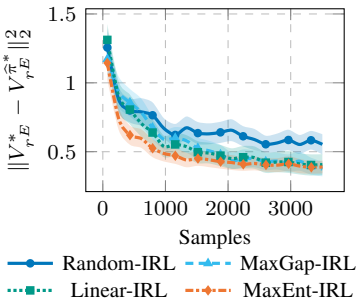

Figure 5. TRAVEL using different IRL algorithms on random MDPs. 200 runs, 98% c.i.

8.3. TRAVEL with Different IRL Algorithms

In this section, we show the performance of TRAVEL paired with different IRL algorithms: MaxGap-IRL, MaxEnt-IRL (Ziebart et al., 2008), Linear-IRL (Abbeel & Ng, 2004), and *Random-IRL*. The Random IRL selects a random reward function from the estimated feasible set of rewards. We compare the algorithms on 200 random generated MDPs. The results in Figure 5 show that MaxGap-IRL, Linear-IRL, and MaxEnt-IRL display a faster convergence rate than Random. This is the expected behavior as these IRL algorithms choose the reward function in the feasible set with a meaningful criterion. However, the curve of Random-IRL shows an improvement, proving that the feasible set shrinks, but it struggles harder to reach a near-zero error as it likely selects less discriminating rewards. This underlines how a reasonable choice of the reward function within the feasible set can have a positive impact on performance.

9. Conclusions

In this paper, we have studied how to efficiently learn a transferable reward from a theoretical perspective. Using the concept of feasible reward set, introduced by Ng & Russell (2000), we have derived novel bounds on the error of the reward function, given an error on the transition model and the expert's policy. We have then obtained similar results on the performance in a target environment using the rewards recovered from a source environment, introducing new simulation lemmas. Based on these findings, we have proposed two algorithms, Uniform Sampling IRL and TRAVEL, which, given a generator model for the source MDP, decide the sampling strategy for querying the generator. These algorithms use an IRL algorithm, decided by the user, as *choice function*. We have derived from the Uniform Sampling IRL a sample complexity bound which, to the best of our knowledge, is the first sample complexity result for the IRL setting. TRAVEL, instead, adapts the sampling strategy to the specific environment at hand. Leveraging this characteristic of the algorithm, we have obtained a problem-dependent bound on its sample complexity. Despite the limitations of the considered setting, we believe that this paper makes a first step towards a better understanding of the theoretical aspects of IRL. Many appealing future research directions arise. One central theoretical question is:

(Q3) *Is Inverse Reinforcement Learning intrinsically more difficult than Reinforcement Learning? Is the sample complexity* $\widetilde{\mathcal{O}}\left(\frac{SA}{(1-\gamma')^2(1-\gamma)^2\epsilon^2}\right)$ *tight?*

We are currently unable to answer. From an algorithmic perspective, our setting limits to tabular MDPs and assumes access to a generative model. Future investigations should include the extension to episode-based interaction and the introduction of function approximation techniques to cope with continuous problems.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In Brodley, C. E. (ed.), *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004. doi: 10.1145/1015330.1015430.

Amin, K., Jiang, N., and Singh, S. P. Repeated inverse reinforcement learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1815–1824, 2017.

Azar, M. G., Munos, R., and Kappen, B. On the sample complexity of reinforcement learning with a generative model. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

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.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Boularias, A., Kober, J., and Peters, J. Relative entropy inverse reinforcement learning. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 182–189, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings.

Chatzigeorgiou, I. Bounds on the lambert function and their application to the outage analysis of user cooperation. *IEEE Communications Letters*, 17(8):1505–1508, 2013.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. *CoRR*, abs/1710.11248, 2017.

Hayat, A., Singh, U., and Namboodiri, V. P. Inforl: Interpretable reinforcement learning using information maximization. *CoRR*, abs/1905.10404, 2019.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4565–4573, 2016.

Judah, K., Fern, A., and Dietterich, T. G. Active imitation learning via reduction to I.I.D. active learning. In de Freitas, N. and Murphy, K. P. (eds.), *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pp. 428–437. AUAI Press, 2012.

Juozapaitis, Z., Koul, A., Fern, A., Erwig, M., and Doshi-Velez, F. Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI Workshop on Explainable Artificial Intelligence*, 2019.

Kakade, S. M. and Langford, J. Approximately optimal approximate reinforcement learning. In Sammut, C. and Hoffmann, A. G. (eds.), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pp. 267–274. Morgan Kaufmann, 2002.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Levine, S., Popovic, Z., and Koltun, V. Nonlinear inverse reinforcement learning with gaussian processes. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. C. N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pp. 19–27, 2011.

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. *Machine Learning*, pp. 1–36, 2021.

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 Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 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.

Ng, A. Y. and Russell, S. J. Algorithms for inverse reinforcement learning. In Langley, P. (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pp. 663–670. Morgan Kaufmann, 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 Schuurmans, D. and Wellman, M. P. (eds.), *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 1993–1999. AAAI Press, 2016.

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

Ramachandran, D. and Amir, E. Bayesian inverse reinforcement learning. In Veloso, M. M. (ed.), *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pp. 2586–2591, 2007.

Ramponi, G., Drappo, G., and Restelli, M. Inverse reinforcement learning from a gradient-based learner. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a.

Ramponi, G., Likmeta, A., Metelli, A. M., Tirinzoni, A., and Restelli, M. Truly batch model-free inverse reinforcement learning about multiple intentions. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2359–2369. PMLR, 2020b.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. Maximum margin planning. In Cohen, W. W. and Moore, A. W. (eds.), *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pp. 729–736. ACM, 2006a. doi: 10.1145/1143844.1143936.

Ratliff, N. D., Bradley, D. M., Bagnell, J. A., and Chestnutt, J. E. Boosting structured prediction for imitation learning. In Schölkopf, B., Platt, J. C., and Hofmann, T. (eds.), *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pp. 1153–1160. MIT Press, 2006b.

Rockafellar, R. T. and Wets, R. J.-B. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In Teh, Y. W. and Titterington, D. M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pp. 661–668. JMLR.org, 2010.

Ross, S., Gordon, G. J., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Gordon, G. J., Dunson, D. B., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pp. 627–635. JMLR.org, 2011.

Russell, J. and Santos, E. Explaining reward functions in markov decision processes. In Barták, R. and Brawner, K. W. (eds.), *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019*, pp. 56–61. AAAI Press, 2019.

Sidford, A., Wang, M., Wu, X., Yang, L., and Ye, Y. Near-optimal time and sample complexities for solving markov decision processes with a generative model. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5192–5202, 2018.

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., Barto, A. G., et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Zanette, A., Kochenderfer, M. J., and Brunskill, E. Almost horizon-free structure-aware best policy identification with a generative model. In Wallach, H. M., Larochelle,

H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5626–5635, 2019.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In Fox, D. and Gomes, C. P. (eds.), *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pp. 1433–1438. AAAI Press, 2008.

Ziebart, B. D., Bagnell, J. A., and Dey, A. K. Modeling interaction via the principle of maximum causal entropy. In Fürnkranz, J. and Joachims, T. (eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pp. 1255–1262. Omnipress, 2010.