Cooperative Inverse Reinforcement Learning

Dylan Hadfield-Menell **Anca Dragan** **Pieter Abbeel** **Stuart Russell**

Abstract

For an autonomous system to be helpful to humans and to pose no unwarranted risks, it needs to align its values with those of the humans in its environment in such a way that its actions contribute to the maximization of value for the humans. We propose a formal definition of the value alignment problem as *cooperative inverse reinforcement learning* (CIRL). A CIRL problem is a cooperative, partial-information game with two agents, human and robot; both are rewarded according to the human's reward function, but the robot does not initially know what this is. In contrast to classical IRL, where the human is assumed to act optimally in isolation, optimal CIRL solutions may include active teaching, active learning, and communicative actions. We show concrete examples where the optimal CIRL solution would be suboptimal for a human acting alone. We also show that under certain conditions CIRL problems can be reduced to POMDPs, and we derive a CIRL algorithm that finds optimal teaching strategies when the robot is assumed to be running an IRL algorithm.

1 INTRODUCTION

"*If we use, to achieve our purposes, a mechanical agency with whose operation we cannot interfere effectively . . . we had better be quite sure that the purpose put into the machine is the purpose which we really desire.*" So wrote Norbert Wiener (1960) in one of the earliest explanations of the problems that arise when autonomous systems operate with objectives that differ from those of humans. This *value alignment* problem is not trivial to solve; humans are prone to mis-stating their objectives, as King Midas found out. Russell & Norvig (2010) give the example of specifying a reward function for a vacuum robot: if we reward the action of cleaning up dirt, which seems reasonable, the robot repeatedly dumps and cleans up the same dirt to maximize its reward. Bostrom (2014) has initiated a broader debate on the long-term implications of this issue. We view the problem as important also in the short term, as a central issue in human–robot interaction (HRI), self-driving cars, personal digital assistants, etc.

Value alignment problems are not unique to artificial systems. Economic systems often involve multiple agents with distinct objectives for whom incentive structures must be designed. Kerr (1975) explains the problems of value misalignment in this context. Our work has strong connections to the *principal–agent* model in game theory, as discussed in Section 2.

Previous work on *inverse reinforcement learning* or IRL (Russell, 1998; Ng & Russell, 2000; Abbeel & Ng, 2004) does address the problem of an AI system—the "robot"—acquiring a reward function from observation of another agent, the "human". It assumes that the human is behaving (approximately) optimally, that the robot is a passive observer during learning, and that, after the learning phase, the robot will adopt the learned reward function as its own. These assumptions strongly restrict the kinds of learning that can take place and omit the need for the robot to make decisions for the benefit of the human, with whom it shares the environment, rather than itself.

Cooperative inverse reinforcement learning. Our main contribution is a formulation of the value alignment problem as *cooperative inverse reinforcement learning* (CIRL). A CIRL problem is a two-player game with partial information, with a given distribution over the reward function (the "private type," in game-speak) of the human. The "human", $\mathbf{H}$, observes the reward function, while the "robot", $\mathbf{R}$, does not; but the robot's payoff is exactly the human's actual reward. Optimal solutions to this game maximize human reward and may involve active instruction by the human and active learning by the robot. As an example, suppose $\mathbf{H}$ has preferences over paperclips and staples. The relative preference for paperclips is described by $\theta \in [0, 1]$. For a state (p, q) with p paperclips and q staples, the reward for both actors is $R((p, q); \theta) = \theta p + (1 - \theta)q$. $\mathbf{H}$

knows θ exactly, but $\mathbf{R}$ has only a prior distribution. As the game unfolds, $\mathbf{R}$ updates the prior based on observation of $\mathbf{H}$'s choices and can also contribute to paperclip and staple production. We show in Section 3 that $\mathbf{H}$, anticipating this, may choose an "instructive" action rather than one that demonstrates "optimal" behavior. We argue in Section 2 that other, related formalisms capture some aspects of this framework but none captures all.

Reduction to POMDP and Sufficient Statistics. The behaviors of human and robot are specified by a pair of policies $(\pi^{\mathbf{H}}, \pi^{\mathbf{R}})$, each depending, in general, on the history of observations and actions. A policy pair yields an expected sum of rewards for each player and is a *Nash equilibrium* if neither actor has an incentive to deviate. In CIRL, the reward function is shared so there is well-defined optimal Nash equilibrium that maximizes value.[1] In Section 3 we show that the problem of computing an optimal policy pair can be reduced to solving a (single-actor) POMDP. The impact of this result is that the class of policy pairs that use $\mathbf{R}$'s *belief* about θ as a sufficient statistic contains an optimal policy pair; moreover, the complexity bound is reduced by an exponential factor compared to general DEC-POMDPS, which are NEXP-hard (Bernstein et al., 2000).

Formal Model of Apprenticeship Learning. In Section 3.3 we model apprenticeship learning (Abbeel & Ng, 2004) as a two-phase CIRL game. In the first phase, the learning phase, both $\mathbf{H}$ and $\mathbf{R}$ can take actions and this lets $\mathbf{R}$ learn about θ. In the second phase, the deployment phase, $\mathbf{R}$ uses what it learned to maximize reward (without supervision from $\mathbf{H}$). We analyze this model to show that classic IRL is a key component of $\mathbf{R}$'s optimal policy under the assumption that $\mathbf{H}$ is an "expert" acting optimally *in isolation* (i.e., assuming no robot exists). As noted above, the robot's presence may lead to "teaching," i.e., deviations by $\mathbf{H}$ from this expert policy. We introduce a somewhat practical algorithm that approximately computes $\mathbf{H}$'s best response to $\mathbf{R}$ when $\mathbf{R}$ uses IRL to track beliefs and rewards are linear in θ and state features. Section 4 compares the results obtained by experts and teachers in a grid-world example and provides empirical confirmation that teaching is better. Thus, designers of apprenticeship learning systems should *expect* users to violate the assumption of expert demonstrations in order to better communicate reward.

2 RELATED WORK

Our proposed model shares aspects with a variety of existing models. We divide the related work into three categories: inverse reinforcement learning, human-aware agents, and principal–agent models.

[1] A coordination problem arises if there is more than one optimal policy pair; we defer this issue to future work.

2.1 Inverse Reinforcement Learning

Ng & Russell (2000) define *inverse reinforcement learning* (IRL) as follows:

"**Given** measurements of an [actor]'s behavior over time. ... **Determine** the reward function being optimized."

The key assumption IRL makes is that the observed behavior is optimal in the sense that the observed trajectory maximizes the sum of rewards. We call this the *demonstration-by-expert* (DBE) assumption. Note that in a CIRL game, this may be *suboptimal* behavior, as $\mathbf{H}$ may choose to accept less reward on a particular action in order to convey more information to $\mathbf{R}$. In CIRL the DBE assumption would fix $\mathbf{H}$'s policy. As a result, many IRL algorithms can be derived as state estimation for a best response to different $\pi^{\mathbf{H}}$.

Ng & Russell (2000), Abbeel & Ng (2004), and Ratliff et al. (2006) compute constraints that characterize the set of reward functions so that the observed behavior maximizes reward. In general, there will be many reward functions consistent with this constraint. They use a max-margin heuristic to select a single reward function from this set as their estimate. In CIRL, the constraints they compute characterize $\mathbf{R}$'s belief about θ under the DBE assumption.

Ramachandran & Amir (2007) and Ziebart et al. (2008) consider the case where $\pi^{\mathbf{H}}$ is "noisily expert," i.e., $\pi^{\mathbf{H}}$is a Boltzmann distribution where actions or trajectories are selected in proportion to the exponent of their value. Ramachandran & Amir (2007) adopt a Bayesian approach and place an explicit prior on rewards. Ziebart et al. (2008) places a prior on reward functions indirectly by assuming a uniform prior over trajectories. In our model, these assumptions are variations of DBE and both implement state estimation for a best response to the appropriate fixed $\mathbf{H}$.

Natarajan et al. (2010) introduce an extension to IRL where $\mathbf{R}$ observes multiple actors that cooperate to maximize a common reward function. This is a different type of cooperation than we consider, as the reward function is common knowledge and $\mathbf{R}$ is a passive observer. Waugh et al. (2011) and Kuleshov & Schrijvers (2015) consider the problem of inferring payoffs from observed behavior in a general (i.e., non-cooperative) game given observed behavior. It would be interesting to consider an analogous extension to CIRL, akin to mechanism design, in which $\mathbf{R}$ tries to maximize collective utility for a group of $\mathbf{H}$s that may have competing objectives.

Fern et al. (2014) consider a *hidden-goal* MDP, a special case of a POMDP where the goal is an unobserved part of the state. This can be considered a special case of CIRL, where θ encodes a particular goal state. The frameworks share the idea that $\mathbf{R}$ helps $\mathbf{H}$. The key difference between the models lies in the treatment of the human (the agent in their terminology). They model the human as part of the environment. In contrast, we treat $\mathbf{H}$ as an actor in a

decision problem that both actors collectively solve.

2.2 Active Learning and Optimal Teaching

Lopes et al. (2009) and Ross et al. (2010) describe *interactive* procedures where artificial agents learn from experts. In Lopes et al. (2009) the robot picks states and an expert gives an expert demonstration in that state. Ross et al. (2010) repeatedly train a policy to imitate a supervised set of examples and then execute it. For each state in the execution, they query an expert for the optimal action. Both of these can be modelled as turn-based CIRL problems where only $\mathbf{R}$'s actions affect the state, and $\mathbf{H}$'s actions serve as discrete labels that $\mathbf{R}$ observes.

The computational study of optimal teaching analyzes the complexity of teaching a concept to a learner (Balbach & Zeugmann, 2009). Goldman et al. (1993) and Goldman & Kearns (1995) consider the problem of determinining optimal teacher/learner pairs and introduce an analogue to VC-dimension that measures the difficulty of a teaching problem. Their treatment of the teacher as an actor shares a strong similarity with CIRL's treatment of $\mathbf{H}$. The key difference is that learning in their formulation is the *objective* of optimal behavior, while in CIRL, it is a consequence of optimal behavior.

Cakmak & Lopes (2012) consider an application of optimal teaching where the goal is to teach the learner the reward function for an MDP. The teacher gets to pick initial states from which an expert executes the reward-maximizing trajectory. They assume the learner uses IRL to infer the reward function and pick initial states to minimize the learner's uncertainty. In CIRL, this approach can be characterized as an approximate algorithm for a highly restricted $\mathbf{H}$ that greedily minimizes the entropy of $\mathbf{R}$'s belief.

There are also several models that deal with the generation of informative actions. Dragan & Srinivasa (2013) give an algorithm that generates motion that is informative of the robot's goal to an observer. Golland et al. (2010) present an algorithm to generate language that is informative of a specific object being referenced in a cluttered scene. Tenenbaum & Griffiths (2001) analyze a model for speech generation where the speaker's goal is to help the listener infer the correct hypothesis. All of these approaches model the observer's inference process and compute actions (motion or utterances) that maximize the probability an observer infers the correct hypothesis or goal. Our approximate solution to CIRL is analogous to these approaches, in that we compute actions that are informative of the correct reward function.

2.3 Principal–agent models

Value alignment problems are not intrinsic to artificial agents. Kerr (1975) describes a wide variety of misaligned incentives in the aptly titled "On the folly of rewarding A, while hoping for B." In economics, this is known as the principal–agent problem: the principal (e.g., the employer) specifies incentives so that an agent (e.g., the employee) maximizes the principal's profit (Jensen & Meckling, 1976).

Principal–agent models study the problem of generating appropriate incentives in a non-cooperative setting with asymmetric information. In this setting, misalignment arises because the agents that economists model are people and intrinsically have their own desires. In AI, misalignment arises entirely from the information asymmetry between the principal and the agent; if we could characterize the correct reward function, we could program it into an artificial agent. Gibbons (1998) provides a useful survey of principal–agent models and their applications. Holmstrom & Milgrom (1987) gives structural results on optimal incentive schemes in linear principal–agent models.

From the perspective of AI research, one of the most interesting lines of research in this literature studies the impacts of distorted incentives. Holmstrom & Milgrom (1991) develop a multi-task model where some tasks are more easily measured and rewarded than others. The key result shows that incentives for the more precisely measured tasks should be reduced to avoid diverting too much effort from poorly measured tasks.

3 COOPERATIVE INVERSE REINFORCEMENT LEARNING

We now introduce the CIRL formulation, show a reduction to a Coordination-POMDP of lower complexity than generic Dec-POMDPs, and characterize Apprenticeship Learning as a simplification of CIRL.

3.1 CIRL Formulation

Definition 1. *A* cooperative inverse reinforcement learning *(*CIRL*) game is a two-player* Dec-POMDP*, M, between a human or principal, $\mathbf{H}$, and a robot or agent, $\mathbf{R}$. The game is described by a tuple, $M = \langle \mathcal{S}, \Theta, P_0, \mathcal{A}^{\mathbf{H}}, \mathcal{A}^{\mathbf{R}}, P_T, H, \gamma, R \rangle$, with the following definitions:*

- $\mathcal{S}$ *a set of world states: $s \in \mathcal{S}$.*
- Θ *a set of possible static reward parameters, only observed by $\mathbf{H}$: $\theta \in \Theta$.*
- P_0 *a distribution over the initial state, represented as tuples: $P_0(s_0, \theta)$*
- $\mathcal{A}^{\mathbf{H}}$ *a set of actions for $\mathbf{H}$: $a^{\mathbf{H}} \in \mathcal{A}^{\mathbf{H}}$.*
- $\mathcal{A}^{\mathbf{R}}$ *a set of actions for $\mathbf{R}$: $a^{\mathbf{R}} \in \mathcal{A}^{\mathbf{R}}$.*

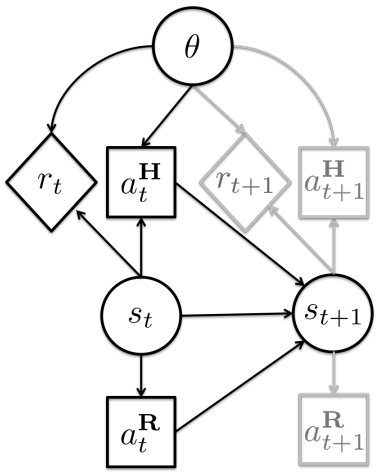

Figure 1: An influence diagram for a single time slice of a CIRL game. Diamonds are reward nodes, circles are random variables, and squares indicate decision nodes for the agents. Both actors receive the same reward, $r_t = R(s_t; \theta)$, but only $\mathbf{H}$ can condition its action selection on the reward parameters θ. $\mathbf{R}$ is forced to infer θ from $\mathbf{H}$'s actions.

P_T *a conditional distribution on the next world state, given previous state and action for both agents:* $P_T(s'|s, a^{\mathbf{H}}, a^{\mathbf{R}})$.
H *a finite horizon for the problem.*
γ *a discount factor:* $\gamma \in [0, 1]$.
R *a parameterized reward function that maps world states, joint actions, and reward parameters to real numbers.* $R : \mathcal{S} \times \mathcal{A}^{\mathbf{H}} \times \mathcal{A}^{\mathbf{R}} \times \Theta \to \mathbb{R}$.

We write the reward for a state–parameter pair as $R(s, a^{\mathbf{H}}, a^{\mathbf{R}}; \theta)$ to distinguish the static reward parameters θ from the changing world state s.

The game proceeds as follows. First, the initial state, a tuple (s, θ), is sampled from P_0. $\mathbf{H}$ observes θ. This parameter represents the human's internal reward function. This observation models that only the human knows the reward function, while both agents know a prior distribution over possible reward functions. At each timestep t, $\mathbf{H}$ and $\mathbf{R}$ observe the current state s_t and select their actions $a_t^{\mathbf{H}}, a_t^{\mathbf{R}}$. Both actors receive reward $r_t = R(s_t, a_t^{\mathbf{H}}, a_t^{\mathbf{R}}; \theta)$ and observe each other's action selection. A state for the next timestep is sampled from the transition distribution, $s_{t+1} \sim P_T(s'|s_t, a_t^{\mathbf{H}}, a_t^{\mathbf{R}})$, and the process repeats. Figure 1 shows an influence diagram for one round of the game.

Behavior in a CIRL problem is defined by a pair of policies, $(\pi^{\mathbf{H}}, \pi^{\mathbf{R}})$, that determine action selection for $\mathbf{H}$ and $\mathbf{R}$ respectively. In general, these policies can be arbitrary functions of their observation histories:

$$\pi^{\mathbf{H}} : \left[\mathcal{A}^{\mathbf{H}} \times \mathcal{A}^{\mathbf{R}} \times \mathcal{S}\right]^* \times \Theta \to \mathcal{A}^{\mathbf{H}} \quad (1)$$

$$\pi^{\mathbf{R}} : \left[\mathcal{A}^{\mathbf{H}} \times \mathcal{A}^{\mathbf{R}} \times \mathcal{S}\right]^* \to \mathcal{A}^{\mathbf{R}} \quad (2)$$

The *value* of a state under a joint policy is the expected sum of discounted rewards under the initial distribution of reward parameters and world states:

$$V^{(\pi^{\mathbf{H}}, \pi^{\mathbf{R}})} = \mathop{\mathbb{E}}_{(s_0,\theta)\sim P_0}\left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t^{\mathbf{H}}, a_t^{\mathbf{R}}) \middle| (\pi^{\mathbf{H}}, \pi^{\mathbf{R}})\right].$$

The optimal joint policy maximizes this value.

Iterated Best Response. A straightforward and efficient approximate algorithm for multi-actor games is to pick an initial policy for all actors and iteratively compute one actor's optimal policy, given a fixed policy for the other actors. We say that such a policy is that actor's *best response* to the other policies.

For example, if $\mathbf{R}$ follows the policy $\pi^{\mathbf{R}}$, then we can define $\mathbf{H}$'s best response as

$$\mathbf{br}(\pi^{\mathbf{R}}) = \mathop{\mathrm{argmax}}_{\pi^{\mathbf{H}}}\left[V^{(\pi^{\mathbf{H}}, \pi^{\mathbf{R}})}\right]. \quad (3)$$

The algorithm iterates through each actor and sets its policy to the best response to the joint policy of the remaining actors. This reaches a fixed point when all policies are mutual best responses: the game-theoretic notion of a *Nash equilibrium* (Gibbons, 1997). Algorithm 1 shows pseudo-code that implements this algorithm.

Algorithm 1 Computing an Equilibrium for CIRL

Define: ITERATEDBESTRESPONSE$(G, (\pi_0^{\mathbf{H}}, \pi_0^{\mathbf{R}}))$
Input: CIRL game G, initial policy pair $(\pi_0^{\mathbf{H}}, \pi_0^{\mathbf{R}})$
$i \leftarrow 0$
$converged \leftarrow False$
while not $converged$ **do**
  /* Compute $\mathbf{H}$'s best response by solving a POMDP where $\mathbf{R}$ is modelled as part of the environment and follows the policy $\pi^{\mathbf{R}}$ */
  $\pi_{i+1}^{\mathbf{H}} \leftarrow \mathbf{br}(\pi_i^{\mathbf{R}})$
  $\pi_{i+1}^{\mathbf{R}} \leftarrow \mathbf{br}(\pi_{i+1}^{\mathbf{H}})$
  /* Check convergence by seeing if $\mathbf{R}$'s policy changed. */
  $converged \leftarrow \pi_{i+1}^{\mathbf{R}} \pi_i^{\mathbf{R}}$
  $i \leftarrow i + 1$
end while
return locally optimal policy pair$\{(\pi_i^{\mathbf{H}}, \pi_i^{\mathbf{R}}$

Next, we show that the problem of computing an optimal policy pair can be reduced to solving a *single-actor* POMDP.

3.2 Reduction to a Coordination POMDP

In this section, we apply the common-information approach of Nayyar et al. (2013) to reduce computing the optimal policy pair in CIRL to optimally solving a related *coordination*-POMDP. The actor in this POMDP is a coordinator that observes all common observations and specifies a policy for each actor. These policies map an actor's private information to an action. Crucially, the structure of CIRL allows this reduction to preserve the size of the (hidden) state space and makes the problem easier to solve, achieving an exponential reduction in complexity compared to arbitrary Dec-POMDPs.

Definition 2. *Let M be a* CIRL *problem between $\mathbf{H}$ and $\mathbf{R}$. We define the corresponding* coordination POMDP *$M_{\mathbf{C}}$ as a* POMDP *where the single actor is a coordinator $\mathbf{C}$. States are tuples of world state and reward parameters: $\mathcal{S}_c = \mathcal{S} \times \Theta$. The intial state distribution places the same distribution on $\mathcal{S} \times \Theta$ as P_0. $\mathbf{C}$'s actions are tuples ($\delta^{\mathbf{H}}$, $a^{\mathbf{R}}$) that specify an action for $\mathbf{R}$ and a decision rule for $\mathbf{H}$ that maps its private information (θ) to an action $\delta^{\mathbf{H}} : \Theta \to \mathcal{A}^{\mathbf{H}}$. $\mathbf{C}$ observes $\mathbf{H}$'s action and the world state. Transitions are defined analogously to those in M.*

A policy in the coordination-POMDP $\pi^{\mathbf{C}}$ maps $\mathbf{H}$'s history of action to an action for $\mathbf{R}$ and a decision rule for $\mathbf{H}$. Note that $\delta^{\mathbf{H}}$ implicitly depends on the world state because $\mathbf{C}$ observes the world state. $M_{\mathbf{C}}$ is equivalent to the original problem M in the sense that, for each joint policy in M, there is a policy of equal value in $M_{\mathbf{C}}$ and vice-versa.

Theorem 1. *Let $M = \langle \mathcal{S}, \Theta, P_0, \mathcal{A}^{\mathbf{H}}, \mathcal{A}^{\mathbf{R}}, P_T, H, R \rangle$ be any* CIRL *problem. Let $M_{\mathbf{C}}$ be the associated coordination* POMDP*. For any strategy pair $(\pi^{\mathbf{H}}, \pi^{\mathbf{R}})$ for M there is an equivalent coordinator policy $\pi^{\mathbf{C}}$.*

Proof. The second component of $\mathbf{C}$'s action is an action for $\mathbf{R}$. $\mathbf{R}$ has no private observations, so clearly $\mathbf{C}$ can simulate any $\pi^{\mathbf{R}}$. $\mathbf{R}$ can also simulate $\mathbf{C}$ and compute the appropriate action.

Turning to $\mathbf{H}$, it is clear that $\mathbf{H}$ can simulate $\pi^{\mathbf{C}}$ and execute the corresponding $\delta^{\mathbf{H}}$. To see that $\mathbf{C}$ can produce an equivalent $\pi^{\mathbf{C}}$ for any $\pi^{\mathbf{H}}$, let h be the action history for $\mathbf{H}$. $\pi^{\mathbf{C}}$ can choose the following decision rule

$$\delta^{\mathbf{H}}(\theta) = \pi^{\mathbf{H}}(\theta; h)$$

to produce the same behavior. □

This reduction lets us show that $\mathbf{R}$'s posterior distribution over θ is a sufficient statistic for optimal behavior. This is $\mathbf{R}$'s belief state and we write it as $b^{\mathbf{R}}$.

Corollary 1. *Let M be a* CIRL *game. There exist optimal policies $(\pi^{\mathbf{H}^*}, \pi^{\mathbf{R}^*})$ that only depend on the current state and $\mathbf{R}$'s belief.*

$$\pi^{\mathbf{H}^*} : \mathcal{S} \times \Delta_\Theta \times \Theta \to \mathcal{A}^{\mathbf{H}}, \quad \pi^{\mathbf{R}^*} : \mathcal{S} \times \Delta_\Theta \to \mathcal{A}^{\mathbf{R}}.$$

Proof. Smallwood & Sondik (1973) showed that an optimal policy in a POMDP only depends on the belief state. $\mathbf{R}$'s belief uniquely determines the belief for $\mathbf{C}$. From this, an appeal to Theorem 1 shows the result. □

Implication. In a general Dec-POMDP, the hidden state for the coordinator-POMDP includes each actor's history of observations. In CIRL, θ is the only private information so we get an exponential decrease in the complexity of the reduced problem. This allows one to apply general POMDP algorithms to compute optimal joint policies in CIRL. It is important to note that the reduced problem may still be very challenging. POMDPs are difficult in their own right and the reduced problem still has a much larger action space. That being said, this reduction is still useful in that it characterizes optimal joint policy computation for CIRL as significantly easier than Dec-POMDPs. Furthermore, this theorem can be used to justify approximate methods (e.g., iterated best response) that only depend on $\mathbf{R}$'s belief state.

3.3 A Formal Model of Apprenticeship Learning

A common paradigm for robot learning from humans is *apprenticeship learning*. In this paradigm, a human gives demonstrations to a robot of a sample task and the robot is asked to imitate it in a subsequent task. In what follows, we formulate apprenticeship learning as turn-based CIRL with a learning phase and a deployment phase. We characterize IRL as the best response to a demonstration-by-expert policy for $\mathbf{H}$. We also show that, in general, it is not an equilibrium policy and so it is a generally suboptimal approach to the problem.

Definition 3. *An* apprenticeship cooperative inverse reinforcement learning *(*ACIRL*) game is a turn-based* CIRL *game with two phases: a learning phase where the human and the robot take turns acting, and a deployment phase, where the robot acts independently.*

As an example, consider an apprenticeship task in the office supply problem from Section 1. Recall that $\mathbf{H}$ and $\mathbf{R}$ can make paperclips and staples and that the unobserved θ describe $\mathbf{H}$'s preference for paperclips vs staples. We model the problem as an ACIRL in which the learning and deployment phase each consist of an individual action.

The world state in this problem is a tuple (p_s, q_s, t) where p_s and q_s respectively represent the number of paperclips and staples $\mathbf{H}$ owns. t is the round number. An action is a tuple (p_a, q_a) that produces p_a paperclips and q_a staples. The human can make 2 items total: $\mathcal{A}^{\mathbf{H}} = \{(0,2), (1,1), (2,0)\}$. The robot has different capabilities. It can make 50 units of each item or it can choose to make 90 of a single item: $\mathcal{A}^{\mathbf{R}} = \{(0,90), (50,50), (90,0)\}$.

We let $\Theta = [0,1]$ and define R so that θ indicates the rela-

tive preference between paperclips and staples:

$$R(s, (p_a, q_q); \theta) = \theta p_a + (1 - \theta) q_a. \quad (4)$$

$\mathbf{R}$'s action is ignored when $t = 0$ and $\mathbf{H}$'s is ignored when $t = 1$. At $t = 2$, the game is over, so we transition to a sink state, $(0, 0, 2)$. Initially, there are no paperclips or staples and we use a uniform prior on θ.

$\mathbf{H}$ only acts in the initial state, so $\pi^{\mathbf{H}}$can be entirely describe by a single decision rule $\delta^{\mathbf{H}} : [0, 1] \to \mathcal{A}^{\mathbf{H}}$. $\mathbf{R}$ only observes one action from $\mathbf{H}$ and so the reachable beliefs are in one-to-one correspondence with $\mathbf{H}$'s actions. This lets us characterize $\mathbf{R}$'s policy as $\pi^{\mathbf{R}} : \mathcal{A}^{\mathbf{H}} \to \mathcal{A}^{\mathbf{R}}$.

In optimal solutions to an ACIRL game, $\mathbf{H}$ uses the learning phase to help the robot figure out what the task is. A result due to Ramachandran & Amir (2007) allows us to characterize $\mathbf{R}$'s optimal policy in the deployment phase: maximize the mean reward function.

Theorem 2. *Let M be an* ACIRL *game. In the deployment phase, the optimal policy for $\mathbf{R}$ maximizes reward in the* MDP *defined by the mean θ.*

Proof. If $\mathbf{R}$ never observes another action from $\mathbf{H}$, then the coordinator POMDP receives no observations. This reduces the problem to solving an MDP under a fixed distribution over reward functions so Theorem 3 from Ramachandran & Amir (2007) shows the result. □

A wide variety of apprenticeship learning tasks assume that demonstrations are given by an expert.

Definition 4. $\mathbf{H}$ *satisfies the* demonstration-by-expert *(*DBE*) assumption in* ACIRL *if it greedily maximizes immediate reward on its turn. If it has multiple turns in a row, it maximizes the sum of rewards collected before the next $\mathbf{R}$ action.*

We use $\mathbf{E}$ to represent actors that satisfy this assumption and $\pi^{\mathbf{E}}$ to represent the corresponding policy.

Theorem 2 allows us to characterize the best reponse for $\mathbf{R}$ under the DBE assumption in ACIRL: use IRL to compute the posterior over θ during the learning phase and then act to maximize reward under the mean θ in the deployment phase. Note that this does not define $\mathbf{R}$'s *behavior* during learning, just its belief.

The DBE assumption in our example assumes that $\mathbf{H}$ maximize reward in the first round:

$$\delta^{\mathbf{E}}(\theta) = \begin{cases} (0, 2) & \theta < 0.5 \\ (1, 1) & \theta = 0.5 \\ (2, 0) & \theta > 0.5 \end{cases}. \quad (5)$$

Let $\theta = 0.49$. $\mathbf{H}$ follows $\delta^{\mathbf{E}}$ and chooses to make 0 paperclips and 2 staples. $\mathbf{R}$ observes this and updates its belief (using $\delta^{\mathbf{E}}$ to define the observation distribution). In this case, we get $b^{\mathbf{R}} = \mathbf{Unif}([0, 0.5))$. Given this belief, $\mathbf{R}$'s maximizes expected reward and chooses to make 0 paperclips and 90 staples. It is not hard to see that the full policy is

$$\mathbf{br}(\delta^{\mathbf{E}})(a^{\mathbf{H}}) = \begin{cases} (0, 90) & a^{\mathbf{H}} = (0, 2) \\ (50, 50) & a^{\mathbf{H}} = (1, 1) \\ (90, 0) & a^{\mathbf{H}} = (2, 0) \end{cases}. \quad (6)$$

Note that when $\theta = 0.49$ $\mathbf{H}$ would prefer $\mathbf{R}$ to choose (50, 50). $\mathbf{H}$ is willing to forgo immediate reward during the demonstration to communicate this to $\mathbf{R}$: the best response to Equation 6 chooses $(1, 1)$ when $\theta = 0.49$. This leads to the following result.

Theorem 3. *Suppose that $\pi^{\mathbf{R}} = \mathbf{br}(\pi^{\mathbf{E}})$. There exist* ACIRL *games where the best-response for $\mathbf{H}$ to $\pi^{\mathbf{R}}$ violates the expert demonstrator assumption. In other words $\mathbf{br}(\mathbf{br}(\pi^{\mathbf{E}})) \neq \pi^{\mathbf{E}}$.*

Proof. Our office supply example gives a counter example that shows the theorem. When $\mathbf{H}$ accounts for $\mathbf{R}$'s actions under $\mathbf{br}(\delta^{\mathbf{E}})$, $\mathbf{H}$ is faced with a choice between 0 paperclips and 92 staples, 51 of each, or 92 paperclips and 0 staples. It is straighforward to show that the optimal decision rule is given by

$$\delta^{\mathbf{H}}(\theta) = \begin{cases} (0, 2) & \theta < \frac{41}{92} \\ (1, 1) & \frac{41}{92} \leq \theta \leq \frac{51}{92} \\ (2, 0) & \theta > \frac{51}{92} \end{cases}.$$

This is distinct from Equation 5 so we conclude the result. □

The key point is that the demonstrator is incentivized to deviate from $\mathbf{R}$'s assumptions.

Remark 1. *We should expect experienced users of apprenticeship learning systems to present demonstrations other than the reward maximizing demonstration.*

This example is simple enough that $\mathbf{R}$ does not need to further adapt its strategy: the best response to novice and experienced users is the same. However, this need not be the case in general and real-world systems need to be robust to both scenarios (see, e.g., Leveson & Turner (1993)).

3.4 Approximate Best Response to Feature Matching

So far we saw that performing IRL to compute a belief over θ and then acting optimally with respect to this belief is the robot's best response to a human *expert*, i.e. a human optimizing reward in isolation. However, we also saw that

the human's best response to this robot policy is *not* to provide expert demonstrations. Here we provide an approximation to ACIRL based on computing what the human's policy should be if the robot is using IRL.

We consider the case where the reward for a state (s, θ) is defined as a linear combination of state features for some feature function ϕ : $R(s;\theta) = \phi(s)^\top \theta$. Standard results from the IRL literature show that policies with the same expected feature counts have the same value (Abbeel & Ng, 2004). In ACIRL, this combines with Theorem 2 to show that the optimal $\pi^{\mathbf{R}}$ under the DBE assumption computes a policy that matches the observed feature counts.

We can use this result to design an approximate best response for $\mathbf{H}$. The main idea is to compute the feature counts that $\mathbf{R}$ expects to see under the true parameters and then select the trajectory that most closely replicates these features. Note that this is generally distinct from the action select by the demonstration-by-expert policy. The goal is to match the expected sum of features under a *distribution* of paths with the sum of features from a *single* path. There are two challenges to overcome in order to turn this into a practical algorithm. We need to be able to efficiently measure how 'similar' two vectors of feature counts are and we need to efficiently compute a demonstration trajectory $\tau^{\mathbf{H}}$ that maximizes this similarity.

The correct measure of feature count similarity is $\mathbf{R}$'s *regret*: the difference between the reward $\mathbf{R}$ would collect if it knew the true θ and the reward $\mathbf{R}$ actually collects using the inferred θ. Unfortunately, this is often very expensive to compute. In this work, we approximate it with an L_2 norm between the difference of the features. We trade-off between this cost and the reward from the trajectory with a parameter η. The optimization we solve is

$$\tau^{\mathbf{H}} \leftarrow \underset{\tau}{\operatorname{argmax}}\ \phi(\tau)^\top \theta - \eta ||\phi_\theta - \phi(\tau)||^2. \qquad (7)$$

where $\phi(\tau) = \sum_{s \in \tau} \phi(s)$ and ϕ_θ is the expected feature counts from following $\pi^{\mathbf{E}}$ under reward parameters θ. Algorithm 2 gives pseudocode to set up the optimization over trajectories. Efficiently solving the optimization is a non-trivial if the demonstrator has to take multiple actions. We do this with an A^* search (Hart et al., 1968). The state in the search is a tuple $(s, \bar{\phi}, t)$ where s is the world state from CIRL, $\bar{\phi}$ is the vector of feature counts observed so far and t is the number of actions taken. The step cost for the search is $-\phi(s)^\top \theta$ at non-terminal states (i.e., states where it is still $\mathbf{H}$'s turn). In terminal states (i.e., states where it is $\mathbf{R}$'s turn) the step cost is weighted feature similarity $\eta\, ||\phi_\theta - \bar{\phi}||^2$.

Algorithm 2 An Approximate Best Response to a Feature Matching $\mathbf{R}$

Define: MATCHFEATURES$(M, \theta, D_\phi, \eta)$
Input: ACIRL game M with state features $\phi(s)$ and linear rewards, the true reward parameters θ, a distance function that measure feature similarity D, η a real number that controls the tradeoff between reward maximization and feature matching.
Solve an MDP to compute the expert policy
$\pi^{\mathbf{E}} \leftarrow$SOLVE$\left(M_\theta^{\mathbf{H}}\right)$
compute the features $\mathbf{R}$ expects under θ
$\phi_\theta \leftarrow \mathbb{E}\left[\sum_t \phi(s_t) \middle| \pi^{\mathbf{E}}\right]$
$L_\theta \leftarrow (\lambda\tau \rightarrow D_\phi(\phi_\theta, \phi(\tau)))$
$\tau^{\mathbf{H}} \leftarrow \underset{\tau}{\operatorname{argmax}}\ \phi(\tau)^\top \theta - \eta L_\theta(\tau))$
return demonstration trajectory $\tau^{\mathbf{H}}$

4 Experiments

In this section, we present an experiment in an cooperative apprenticeship problem for 2D mobile robot navigation. We show the benefit of CIRL by comparing the approximation from Section 3.3 to IRL. Specifically, we examine the case where $\mathbf{R}$ makes the DBE assumption (i.e., $\mathbf{R}$ is implemented with IRL) and measure the value of different policies for $\mathbf{H}$: *expert*, which matches the IRL assumption, and *best-responder*, which computes the best response to IRL. Before giving the details of our experiment, we define our problem domain.

4.1 Cooperative Learning for Mobile Robot Navigation

Our experimental domain is a 2D navigation problem on a discrete grid. In the learning phase of the game, $\mathbf{H}$ teleoperates a trajectory while $\mathbf{R}$ observes. In the deployment phase, $\mathbf{R}$ controls the robot and attempts to maximize reward. We use a finite horizon H, and let the first $\frac{H}{2}$ timesteps be the learning phase.

There are N_ϕ state features defined as radial basis functions where the centers are common knowledge. Rewards are linear in these features and θ. The intial world state is in the middle of the map. We use a uniform distribution on $[-1, 1]^{N_\phi}$ for the prior on θ. Actions move in one of the four cardinal directions $\{N, S, E, W\}$ and there is an additional no-op $\emptyset$ that each actor executes deterministically on the other agent's turn.

If N is the number of grid cells per axis and $\vec{c}$ is the vector of RBF centers, we can define our problem as follows:

- $\mathcal{S} = [0, \ldots, N-1] \times [0, \ldots, N-1] \times [0, \ldots, H]$
- $\Theta = [-1, 1]^{N_\phi}$
- $P_0 = \mathbf{Dirac}(N/2, N/2, 0) \times \mathbf{Unif}([-1, 1]^{N_\phi})$

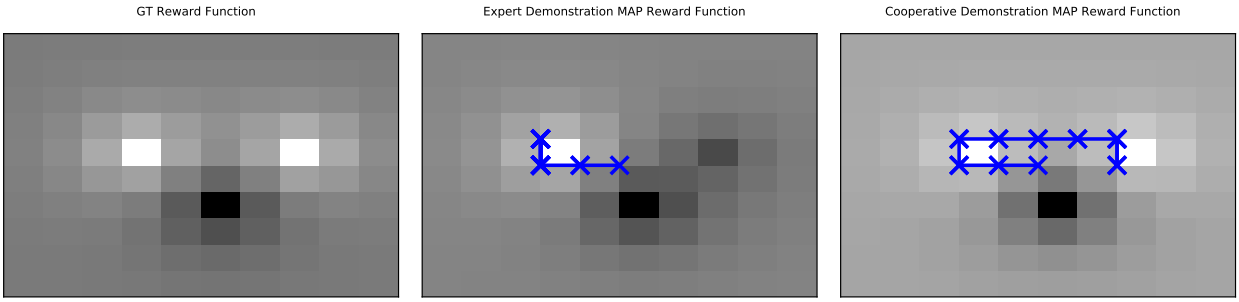

Figure 2: The difference between demonstration-by-expert and communicative demonstration in the mobile robot navigation problem from Section 4. Left: The ground truth reward function. Lighter grid cells indicates areas of higher reward. Middle: The demonstration trajectory generated by the expert policy, superimposed on the maximum a-posteriori reward function the robot infers. Notice that the path remains at a single point of high reward for several steps. The robot successfully learns where the maximum reward is, but little else. Right: The demonstration generated by Algorithm 2 superimposed on the maximum a-posteriori reward function that the robot infers. The demonstration highlights both points of high reward and so the robot learns a better estimate of the reward.

- $\mathcal{A}^{\mathbf{H}} = \mathcal{A}^{\mathbf{R}} = \{N, S, E, W, \emptyset\}$
- P_T moves in the direction indicated deterministically, based on whose turn it is. At the start of the deployment phase ($t = \frac{H}{2}$), the next state is sampled uniformly at random.
- H is the horizon
- $\gamma = 1$
- $R(s, a^{\mathbf{H}}, a^{\mathbf{R}}; \theta) = \phi(s; \vec{c})^\top \theta$

Figure 2 shows an example comparison between demonstration-by-expert and the approximate best response policy in Section 3.4. The leftmost image is the ground truth reward function. Next to it are demonstration trajectories produce by these two policies. Each path is superimposed on the maximum a-posteriori reward function the robot infers from the demonstration. We can see that the demonstration-by-expert policy immediately goes to the highest reward and stays there. In contrast, the best response policy moves to both areas of high reward. The robot reward function the robot infers from the best response demonstration is much more representative of the true reward function, when compared with the reward function it infers from demonstration-by-expert.

4.2 Demonstration-by-Expert vs Best Responder

Hypothesis. When $\mathbf{R}$ plays an IRL algorithm that matches features, $\mathbf{H}$ prefers $\hat{\mathbf{br}}(\pi^{\mathbf{R}})$ to $\pi^{\mathbf{E}}$: the best response policy will significantly outperform the demonstration-by-expert policy.

Manipulated Variables: Our experiment consists of 2 factors: **H-*policy*** and ***num-features***.

We make the assumption that $\mathbf{R}$ uses an IRL algorithm to compute its estimate of θ during learning and maximizes reward under this estimate during deployment. We use Maximum-Entropy IRL (Ziebart et al., 2008) to implement $\mathbf{R}$'s policy. W

H-*policy* varies $\mathbf{H}$'s strategy $\pi^{\mathbf{H}}$and has two levels: demonstration-by-expert and best-responder. In the demonstration-by-expert level $\mathbf{H}$ adheres to the DBE assumption and maximizes reward during the demonstration. In the best-responder level $\mathbf{H}$ uses Algorithm 2 to compute an (approximate) best response to $\pi^{\mathbf{R}}$. The trade-off between reward and communication η is set by cross-validation before the game begins.

The ***num-features*** factor varies the dimensionality of ϕ. We consider two levels for this factor: 3 features and 10 features. We do this to test whether and how the difference between experts and best-responders is affected by dimensionality.

We use a factorial design across these factors. This leads to 4 distinct conditions.

We test each condition against a random sample of $N = 500$ different reward parameters. We use a within-subjects design with respect to the the **H-policy** factor so the same reward parameters are tested for both levels. We use a between-subjects design for the ***num-features*** factor, as this changes the dimensionality of the reward parameters. We use $\eta = 10^{-4}$ when ***num-features***=3 and $\eta = 10^{-3}$ when ***num-features***=10.

Dependent Measures: We use the regret with respect to a fully-observed setting where the robot knows the ground truth θ as a measure of performance. We let $\hat{\theta}$ be the robot's estimate of the reward parameters and let θ_{GT} be the ground truth reward parameters. The primary measure is the sub-optimality of robot's policy reward: the difference between the value of the policy that maximizes $\hat{\theta}$ and the value of the policy that maximizes θ_{GT}. We also use two secondary measures. The first is the KL-divergence between the maximum-entropy trajectory distribution in-

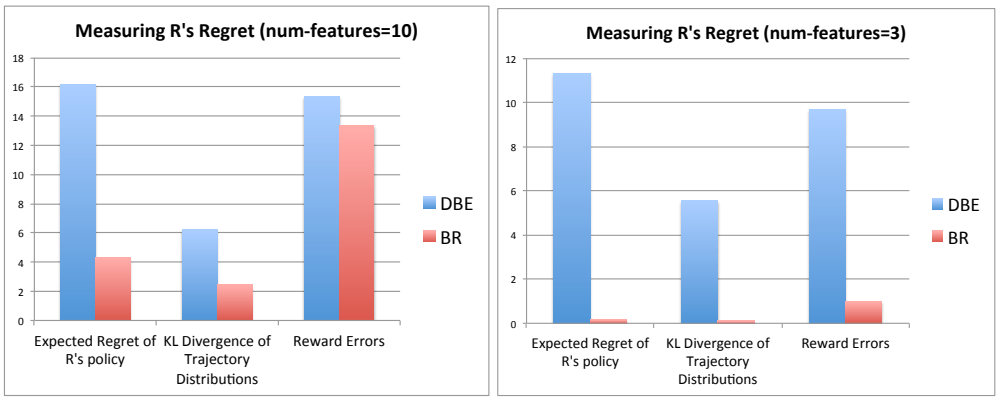

Figure 3: Bar graph of the dependent measures from our experiment. We can see that there is a substantial gap between the regret incurred by the approximate best response given in Algorithm 2 (**br**) is substantially less than that incurred by the demonstration-by-expert (DBE) policy. We can see that the benefit is larger when the number of features is smaller. This is because the number of features governs the dimensionality of θ, thus informative demonstrations can have a larger effect on $\mathbf{R}$'s belief. It is interested that in the ***num-features***=10 condition the reward function $\mathbf{R}$determines is quite wrong (rightmost bar) but the regret of the optimal policy is quite low. This shows that $\mathbf{H}$has a strong incentive to use a policy other than demonstration-by-expert.

duced by $\hat{\theta}$ and the maximum-entropy trajectory distribution induced by θ. Finally, we use the ℓ_2-norm between the vector or rewards defined by $\hat{\theta}$ and the vector induced by θ.

Results: Figure 3 shows the dependent measures from our experiment. We are able to confirm our hypothesis that the demonstration-by-expert level of the **H-*policy*** results is substantially higher regret than the best-response policy. The size of the gap is especially large. One particularly interesting observation is that even when $\mathbf{R}$'s inferred reward function is inaccuracte the best-response policy still results in low regret with respect to the value of the policy pair.

5 CONCLUSION AND FUTURE WORK

In this work, we presented a game-theoretic model for cooperative learning, CIRL. Key to this model is that the robot *knows* that it is in a shared environment and is attempting to maximize the human's reward (as opposed to estimating the human's reward function and adopting it as its own). This leads to cooperative learning behavior and provides a framework in which to design HRI algorithms and analyze the incentives of both actors in a learning environment.

We reduced the problem of computing an optimal policy pair to solving a POMDP. This is a useful theoretical tool and can be used to design new algorithms, but it is clear that optimal policy pairs are only part of the story. In particular, when it performs a centralized computation, the reduction assumes that we can effectively program both actors to follow a set coordination policy. This may not be feasible in reality, although it may nonetheless be helpful in training humans to be better teachers. An important avenue for future research will be to consider the problem of *equilibrium acquisition*: the process by which two independent actors arrive at an equilibrium pair of policies. Returning to Wiener's warning, we believe that the best solution is not to put a specific purpose into the machine at all, but instead to design machines that provably converge to the right purpose as they go along.

References

Abbeel, P and Ng, A. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

Balbach, F and Zeugmann, T. Recent developments in algorithmic teaching. In *Language and Automata Theory and Applications*. Springer, 2009.

Bernstein, D, Zilberstein, S, and Immerman, N. The complexity of decentralized control of Markov decision processes. In *UAI*, 2000.

Bostrom, N. *Superintelligence: Paths, dangers, strategies*. Oxford, 2014.

Cakmak, M and Lopes, M. Algorithmic and human teaching of sequential decision tasks. In *AAAI*, 2012.

Dragan, A and Srinivasa, S. Generating legible motion. In *Robotics: Science and Systems*, 2013.

Fern, A, Natarajan, S, Judah, K, and Tadepalli, P. A decision-theoretic model of assistance. *JAIR*, 50(1):71–104, 2014.

Gibbons, R. *An introduction to applicable game theory*. National Bureau of Economic Research, 1997.

Gibbons, R. Incentives in organizations. Technical report, National Bureau of Economic Research, 1998.

Goldman, S and Kearns, M. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

Goldman, S, Rivest, R, and Schapire, R. Learning binary relations and total orders. *SIAM Journal on Computing*, 22(5):1006–1034, 1993.

Golland, D, Liang, P, and Klein, D. A game-theoretic approach to generating spatial descriptions. In *EMNLP*, pp. 410–419, 2010.

Hart, P, Nilsson, N, and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 100–107, 1968.

Holmstrom, B and Milgrom, P. Aggregation and linearity in the provision of intertemporal incentives. *Econometrica*, pp. 303–328, 1987.

Holmstrom, B and Milgrom, P. Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design. *Journal of Law, Economics, & Organization*, 7: 24–52, 1991.

Jensen, M and Meckling, W. Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3(4):305–360, 1976.

Kerr, S. On the folly of rewarding A, while hoping for B. *Academy of Management Journal*, 18(4):769–783, 1975.

Kuleshov, V and Schrijvers, O. Inverse game theory. *Web and Internet Economics*, 2015.

Leveson, N and Turner, C. An investigation of the Therac-25 accidents. *IEEE Computer*, 26(7):18–41, 1993.

Lopes, M, Melo, F, and Montesano, L. Active learning for reward estimation in inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009.

Natarajan, S, Kunapuli, G, Judah, K, Tadepalli, P, and Kersting, Kand Shavlik, J. Multi-agent inverse reinforcement learning. In *Int'l Conference on Machine Learning and Applications*, 2010.

Nayyar, A, Mahajan, A, and Teneketzis, D. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013.

Ng, A and Russell, S. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

Ramachandran, D and Amir, E. Bayesian inverse reinforcement learning. In *IJCAI*, 2007.

Ratliff, N, Bagnell, J, and Zinkevich, M. Maximum margin planning. In *ICML*, 2006.

Ross, S, Gordon, G, and Bagnell, J. A reduction of imitation learning and structured prediction to no-regret online learning. *arXiv preprint arXiv:1011.0686*, 2010.

Russell, S. and Norvig, P. *Artificial Intelligence*. Pearson, 2010.

Russell, Stuart J. Learning agents for uncertain environments (extended abstract). In *COLT*, 1998.

Smallwood, R and Sondik, E. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

Tenenbaum, J and Griffiths, T. The rational basis of representativeness. In *CogSci*, 2001.

Waugh, K, Ziebart, B, and Bagnell, J. Computational rationalization: The inverse equilibrium problem. In *ICML*, 2011.

Wiener, N. Some moral and technical consequences of automation. *Science*, 131, 1960.

Ziebart, B, Maas, A, Bagnell, J, and Dey, A. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.