# Enabling Environment Design via Active Indirect Elicitation

**Haoqi Zhang** and **David C. Parkes**
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
{hq, parkes}@eecs.harvard.edu

## Abstract

Many situations arise in which an interested party wishes to affect the decisions of an agent; e.g., a teacher that seeks to promote particular study habits, a Web 2.0 site that seeks to encourage users to contribute content, or an online retailer that seeks to encourage consumers to write reviews. In the problem of *environment design*, one assumes an interested party who is able to alter limited aspects of the environment for the purpose of promoting desirable behaviors. A critical aspect of environment design is understanding preferences, but by assumption direct queries are unavailable. We work in the inverse reinforcement learning framework, adopting here the idea of active indirect preference elicitation to learn the reward function of the agent by observing behavior in response to incentives. We show that the process is convergent and obtain desirable bounds on the number of elicitation rounds. We briefly discuss generalizations of the elicitation method to other forms of environment design, e.g., modifying the state space, transition model, and available actions.

## Introduction

Many situations arise in which an interested party wishes for an agent to behave in a certain way. A teacher wants a student to form effective study habits. A Web 2.0 site wants a user to contribute content. An online retailer wants a customer to make purchases and write reviews on products bought. But often, the agent's actual behavior differs from the behavior desired by the interested party. For one, the agent may have different preferences; e.g., a student may not value getting the right answer as much as the teacher and a user may not derive much value from sharing content. Furthermore, the agent may be limited by personal and environmental constraints; e.g., a student may not know the techniques necessary for solving the problem effectively and a consumer may have trouble finding the product he is looking for. Another possibility is that the agent is being limited by the actions she is allowed to take; e.g. a student may not be given a chance to participate in class or a user is restricted from sharing content by the ISP. The converse problem may also exists, in which an agent may be taking actions that should be restricted but aren't.

Underlying these possible explanations is the view that the *environment* in which the agent is in – by that we mean that which includes states, rewards, available actions, and transition functions – has a direct effect on the agent's behavior. It seems plausible then that if the interested party is able to alter aspects of this environment (by providing incentives, modifying the set of available actions, altering the transition probabilities, and 'landscaping' the physical state space, etc.) that the interested party can indirectly affect the behavior of the agent. Of course, the aspects of the environment that the interested party can alter are limited and such alterations are likely to be costly and may have unexpected or undesirable consequences. Nevertheless, if the interested party can obtain a fairly accurate model of the environment and the agent's preferences, the interested party may be able to make effective changes.

We view this as a class of problems of *environment design.* Following the concept from Zhang and Parkes (2008), we envision a setting where an agent performs a sequence of observable actions in an environment, repeatedly and relatively frequently. An interested party has measurements of the agent's behavior over time, and can modify limited aspects of the environment. The agent may choose to behave differently in the modified environment, but the interested party cannot directly impose actions on an agent. The goal of the interested party is to induce a desired behavior quickly and at a low cost.

A critical aspect of environment design is understanding an agent's preferences, which are often complex and unknown to the interested party. In the preference elicitation literature, this is typically done by asking the agent a series of direct queries (Boutilier et al. 2005; Boutilier, Sandholm, and Shields 2004; Chajewska, Koller, and Parr 2000; Wang and Boutilier 2003), based on which the elicitor places bounds on the agent's utility function. While direct elicitation methods have been successfully applied to settings such as combinatorial auctions (Sandholm and Boutilier 2006) and user interface optimization (Gajos and Weld 2005), we believe this approach would be infeasible and undesirable for environment design. A direct elicitation process is costly here: with interdependent world states it is difficult for the agent to accurately report rewards on individual states, and reporting preferences for policies is difficult given the large number of potential policies, few of which the agent may have considered explicitly. Most importantly, direct elicitation is intrusive and outside the "indirect" spirit of environment design.

In this paper, we provide methods for environment design through an active indirect elicitation approach. Working in the Markov Decision Process framework, we illustrate our methods through the problem of *policy teaching* (Zhang and Parkes 2008), where an interested party is able to associate limited rewards with world states and in the setting considered in this paper *wishes to induce the agent to follow a particular policy*. When agent rewards are known, we present a simple linear program that uses techniques from inverse reinforcement learning (IRL) (Ng and Russell 2000) to associate payments with states to induce a desired policy while minimizing expected cost. When rewards are unknown, we repeat a process of guessing the reward function, providing incentives treating the guess as the actual reward, and observing the resulting policy.

We present an algorithm that iteratively learns the agent's reward function by narrowing down the space of possible agent rewards until the provided incentive induces the desired policy. Our elicitation method allows for many elicitation strategies (in terms of how incentives are provided in seeking to narrow down the space of possible rewards). We discuss possible strategies and provide desirable bounds on the number of elicitation rounds for specific strategies. Furthermore, we consider a variant of the *two-sided slack-maximizing heuristic* from Zhang and Parkes (2008) that is easy to compute and can lead to very few elicitation rounds.

We close with a brief discussion of possible generalizations of this indirect, active elicitation approach to other forms of environment design, considering settings in which the interested party is able to perturb the available actions and transitions in the environment, in addition to adjust an agent's rewards through incentive payments.

## Related work

The idea of policy teaching is inspired by applications of inverse reinforcement learning to apprenticeship learning. Abbeel and Ng (2004) studied this problem by extracting a reward function from a human expert, and used the acquired reward function to govern the behavior of a machine agent. In our work, we cannot redefine the agent's reward function at will; the interested party may only provide incentives to induce the agent to behave according to both the provided incentives and the agent's inherent preferences. Furthermore, in providing incentives, the size of the incentives must be in line with the size of the reward function of the agent. For example, if the agent is buying a car, providing a five dollar discount would be insufficient, but we may nevertheless make this mistake if we do not learn both the shape and size of the agent's reward. In Abbeel and Ng's work, any non-degenerate reward function within the solution space is sufficient, out of which they pick one that generalizes well.

Our active, indirect elicitation method alters the environment (e.g. provides incentives), and actively generates new evidence about the agent's reward function based on its behavior in the modified environment. To our knowledge, this approach has not been previously studied in the literature, and may be useful for learning preferences in a wide range of settings. While indirect elicitation techniques based on the principles of revealed preference are nothing new (see Varian (2003) for a survey), such techniques are typically passive (Chajewska, Koller, and Ormoneit 2001; Ng and Russell 2000); they are applied to observed behaviors within a fixed environment and are unconcerned with generating new evidence from which to make further inferences about the agent's preferences.

In previous work we studied the related problem of *value-based policy teaching*, which differs from that considered here in that the goal of the interested party is to induce the agent to follow a policy that maximizes the total expected value of the interested party (Zhang and Parkes 2008). The problem there is NP-hard, whereas policy teaching to induce a particular policy (as in this paper) can be formulated with a linear program. One advantage of this work is that we have a polynomial time algorithm that can simultaneously compute desired incentive provisions while ensuring elicitation convergence in number of rounds logarithmic in the size of the search space with arbitrarily high probability (neither guarantees were provided in the other work).

# The Model

We model an agent performing a sequential decision task with an infinite horizon Markov Decision Process (MDP) $M = \{S, A, R, P, \gamma\}$, where $S$ is the set of states, $A$ is the set of possible actions, $R : S \rightarrow \mathbb{R}$ is the reward function, $P : S \times A \times S \rightarrow [0, 1]$ is the transition function, and $\gamma$ is the discount factor from $(0, 1)$. Given an MDP, an agent's decision problem is to maximize the expected sum of discounted rewards. We consider the agent's decision as a stationary policy $\pi$, such that $\pi(s)$ is the action the agent executes in state $s$. Given a policy $\pi$, the value function $V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P(s, \pi(s), s') V^\pi(s')$ captures the expected sum of discounted rewards under $\pi$. Similarly, the Q function captures the value of taking an action $a$ and following the policy $\pi$ in future states, such that $Q^\pi(s, a) = R(s) + \gamma \sum_{s' \in S} P(s, a, s') V^\pi(s')$. By Bellman optimality (Puterman 1994), an optimal policy $\pi^*$ chooses actions that maximize the $Q$ function in every state, such that $\pi^*(s) \in \text{argmax}_{a \in A} Q^{\pi^*}(s, a)$.

Having defined the MDP framework, we now consider the policy teaching problem. Given an agent facing an MDP $M = \{S, A, R, P, \gamma\}$, an interested party wishes to provide an incentive function $\Delta : S \rightarrow \mathbb{R}$ to induce a desired target policy $\pi_T$. To capture the limits on the incentives that an interested party can provide, we require that $\Delta$ satisfy the following admissibility definition:

**Definition 1.** An incentive function $\Delta$ is *admissible* with respect to a policy $\pi_T$ if it satisfies the following linear constraints:

$$V_\Delta^{\pi_T}(s) = \Delta(s) + \gamma P_{s,\pi_T(s)} V_\Delta^{\pi_T}, \forall s \qquad \textit{Incentive value.}$$
$$V_\Delta^{\pi_T}(start) \leq D_{max} \qquad \textit{Limited spending.}$$
$$\Delta(s) \geq 0, \forall s \qquad \textit{No punishments.}$$

This notion of admissibility limits the expected discounted sum of provided incentives to $D_{max}$ when the agent performs $\pi_T$ from the start state.[1] Here $V_\Delta^{\pi_T}$ is defined over

[1] Notice that the use of a single start state is without loss of gen-

provided incentives and is analogous to the value function $V^\pi$ defined on rewards. Alternative definitions of admissibility are possible as well, but since our methods are not specific to a particular definition, we will not pursue them here.

We make the following assumptions:

**Assumption 1.** The state and action spaces are finite.

**Assumption 2.** The agent's reward function is bounded in absolute value by $R_{max}$.

**Assumption 3.** The agent can compute $\pi^*$ (i.e., the agent is a planner).

**Assumption 4.** The interested party observes the agent's policy $\pi^*$.

**Assumption 5.** The agent's reward function is state-wise quasilinear; given incentive $\Delta$, the agent plans with respect to $R + \Delta$.

**Assumption 6.** The agent's reward function is persistent.

**Assumption 7.** The agent is *myopically rational*. Given $\Delta$, the agent plans with respect to $R + \Delta$ and does not reason about incentive provisions in future interactions with the interested party.

The planning assumption is quite fundamental to our work. Our tenet is that the agent will adjust, not necessarily immediately, to following a behavior that maximizes its reward (including the incentives that it receives for its behavior) in a perturbed environment. Note that this does not necessarily imply that the agent knows, or understands, its underlying preferences. Rather, the agent behaves in the long-run as an optimal "planner" with respect to the uncertain environment in which it is situated.

Assumptions 3, 6 and 7 are difficult to satisfy, but do not pose major issues if they are only mildly violated, e.g. the agent can almost plan, acts with respect to an almost constant reward, and reasons mostly with respect to the current interaction.[2] Violations of assumption 4 can be handled when the interested party can obtain samples of the agent's action trajectories through the state space (which can then be used to obtain a linear approximation of the agent's reward function, e.g. see Ng and Russell (2000)).

## The case with known rewards

While we don't expect to know the agent's reward function, it is nevertheless instructive to understand how to find the minimal $\Delta$ such that $R + \Delta$ induces the desired policy $\pi_T$ for any reward function $R$.

**Definition 2. Policy teaching with known rewards.** An agent follows an optimal policy $\pi$ with respect to an MDP $M = \{S, A, R, P, \gamma\}$. An interested party observes $\pi$ and has knowledge of the agent's MDP $M$. Find a minimal admissible $\Delta$ to induce the agent to perform some desired policy $\pi_T$ and strictly prefer this to any other policy (or show that no such $\Delta$ is available).

Solving this problem requires finding an admissible incentive mapping to some reward function for which $\pi_T$ is the optimal policy. The space of rewards that correspond to a particular optimal policy contains all reward functions for which the actions of the optimal policy maximize the value of the Q function. One can express this space through a set of linear constraints, referred to as the IRL constraints:

**Definition 3.** Given a policy $\pi$ and $M_{-R} = \{S, A, P, \gamma\}$, let $R \in \text{IRL}^\pi$ denote the space of all reward functions $R$ for which $\pi$ is optimal for the MDP $M = \{S, A, R, P, \gamma\}$.

**Theorem 1.** *(Ng and Russell 2000) Given a policy $\pi$ and $M_{-R} = \{S, A, P, \gamma\}$, $R \in IRL^\pi$ satisfies:*

$$(\mathbf{P}_\pi - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_\pi)^{-\mathbf{1}}\mathbf{R} \succeq \mathbf{0} \quad \forall a \in A \tag{1}$$

The result follows from writing the functions of the MDP in vector form (justified by finite state and actions from Assumption 1) and applying the equality $\mathbf{V}^\pi = (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-\mathbf{1}}\mathbf{R}$. With bounded rewards (Assumption 2), these linear constraints define a $|S|$-dimensional convex polytope (which we refer to as "the IRL space") where the reward function resides. To solve the policy teaching problem with known rewards, we can place IRL constraints on the space of rewards that induce the target policy $\pi_T$, and aim to find the minimal admissible $\Delta$ that maps to a reward in this space.

**Theorem 2.** *The following linear program solves the policy teaching problem in Definition 2:*

$$\min_{\Delta} V_\Delta^{\pi_T}(start) \tag{2}$$

*subject to:*

$$R_T(s) - \Delta(s) = R(s) \qquad \forall s \in S \tag{3}$$

$$(\mathbf{P}_{\pi_\mathbf{T}} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi_\mathbf{T}})^{-\mathbf{1}}\mathbf{R_T} \succeq \epsilon \qquad \forall a \in A \backslash a_1 \tag{4}$$

$$admissible(\Delta) \tag{5}$$

*where $a_1 \equiv \pi_T(s)$ denotes the actions of the target policy.*

The result follows directly from Definition 2 and Theorem 1. The use of a small $\epsilon > 0$ on the right hand side of the IRL constraints imposes a strictness condition on the mapping, such that under $R_T$ the agent strictly prefers the desired policy $\pi_T$ over any other policy with slack at least $\epsilon$. This condition ensures that we map only to rewards for which $\pi_T$ is the unique optimal policy, avoiding scenarios with ties where the agent may choose an alternate policy. When the linear program for $\epsilon > 0$ is infeasible then there are no admissible $\Delta$ that can "$\epsilon$-strictly" induce the desired policy.[3]

erality, since it can be a dummy state whose transitions represent a distribution over possible start states.

[2] In fact, we can also show that while a forward-looking (i.e. strategic) agent may choose to misrepresent its preferences, a simple teaching rule can nevertheless teach the desired policy when the agent is sufficiently patient (and as long as the behavior is "teachable", meaning that it is attainable given the limited incentives available to the interested party).

[3] Note that there exist policies that can't be strictly induced by any reward function. As an example, consider a 2 state MDP with actions `stay` and `move`. A policy which chooses `stay` in

## The case with unknown rewards

In most situations, the interested party will not know the reward function of the agent. For simplicity we drop the requirement for incentives to be minimal and just focus on finding any admissible mapping to the desired policy from the agent's unknown reward:

**Definition 4. Policy teaching with unknown rewards.** An agent follows an optimal policy $\pi$ with respect to an MDP $M = \{S, A, R, P, \gamma\}$. An interested party observes $\pi$ and has knowledge of $M_{-R} = \{S, A, P, \gamma\}$, but does not know the agent's reward. Find an admissible $\Delta$ to induce the agent to perform some desired policy $\pi_T$ and strictly prefer this to any other policy (or show that no such $\Delta$ is available).

While IRL constraints provide bounds on the space of rewards that induce the agent's policy, they do not immediately locate the agent's actual reward within this space. This presents a problem because the particular incentives required to induce the desired policy can depend on the specific reward profile. To overcome this problem, we describe a scheme wherein we narrow the space of potential agent rewards by eliciting additional IRL constraints based on an agent's response to provided incentives.

We begin with IRL constraints from the observed agent policy. Furthermore, since we are only interested in rewards that have admissible mappings to the desired policy, we need only consider rewards $R'$ with an associated admissible $\Delta(R')$ mapping to some reward $R_T$ that $\epsilon$-strictly induces $\pi_T$ (for parameter $\epsilon > 0$). From this set of rewards, we make a guess $\widehat{R}$ at the agent's reward. If our guess is correct, we would expect that providing the agent with incentive $\widehat{\Delta}$ will induce the agent to perform $\pi_T$. If instead the agent performs a policy $\pi' \neq \pi_T$, we know that $\widehat{R}$ must not be the agent's true reward $R$, and since $R + \widehat{\Delta}$ induces $\pi'$, we elicit additional information which will eliminate other points in the space of agent rewards.

Using the observation of the agent's policy $\pi'$ in response to the provided incentive, we can write down an IRL constraint on $R + \widehat{\Delta}$ such that $(R + \widehat{\Delta}) \in \text{IRL}^{\pi'}$:

$$(\mathbf{P}_{\pi'} - \mathbf{P}_{\mathbf{a}})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-\mathbf{1}}(\mathbf{R} + \widehat{\mathbf{\Delta}}) \succeq \mathbf{0} \quad \forall a \in A \qquad (6)$$

We can repeat the process of choosing a reward in the agent's refined IRL space, mapping it to a point in the IRL space of a desired policy, observing the induced agent policy, and adding new constraints if the agent does not behave as desired. With direct queries, responses typically imply just one constraint on the space of possible valuations. In our setting, each set of added IRL constraints describes a polytope whose intersection with the polytope describing the existing IRL space defines the updated space of possible agent rewards. Also, since we are only interested in the agent's reward for the purpose of solving the policy teaching problem, we can stop the elicitation process as soon as an admissible mapping to the desired policy is found, regardless of whether $\widehat{R}$ is indeed the agent's true reward.

both states implies a reward function with equal value for the two states and thus the policy cannot be strictly preferred. When reward functions are generalized to state-action pairs, every policy can be strictly supported by some reward function, e.g. by assigning equal positive rewards to state-action pairs matching the policy and no rewards to all other pairs. This extension can be trivially handled and all results will still apply.

**Algorithm 1** Active indirect elicitation for policy teaching

**Require:** agent policy $\pi$, desired policy $\pi_T$
1: Variables $R$, $R_T$, $\Delta$; constraint set $K = \emptyset$
2: Add $R \in \text{IRL}^{\pi}$, $|R(s)| \leq R_{max} \forall s \in S$ to $K$
3: Add $R_T \in \text{IRL}^{\pi_T}_{strict(\epsilon)}$, $\Delta = R_T - R$ to $K$
4: Add $admissible(\Delta)$ to $K$
5: **loop**
6: Find $\widehat{\Delta}$, $\widehat{R}$, $\widehat{R_T}$ satisfying all constraints in $K$
7: **if** no such values exist **then**
8: return FAILURE {no possible mappings}
9: **else**
10: Provide agent with incentive $\widehat{\Delta}$
11: Observe $\pi'$ with respect to $R' = R^{true} + \Delta$.
12: **if** $\pi' = \pi_T$ **then**
13: return $\widehat{\Delta}$
14: **else**
15: Add $(R + \widehat{\Delta}) \in \text{IRL}^{\pi'}$ to $K$

We adopt the following notation for our algorithm. We denote IRL constraints on reward profile $R$ as $R \in \text{IRL}^{\pi}$, and strict IRL constraints (following Equation 4) over target reward $R_T$ as $R_T \in \text{IRL}^{\pi_T}_{strict(\epsilon)}$. All constraints are added to a constraint set $K$, such that feasible solutions must satisfy all constraints in $K$. An instantiation of a variable $R$ is denoted as $\widehat{R}$. Algorithm 1 gives our elicitation method.

**Theorem 3.** *Algorithm 1 terminates in a finite number of steps with an admissible mapping $\Delta$ to the target policy $\pi_T$, or returns FAILURE if no such mapping exists.*

The proof is omitted in the interest of space. It adopts the minimal slack $\epsilon$ from the strictness condition on mappings to the target policy to bound the number of hypercubes that can fit within the IRL space. The number of elicitation rounds is bounded by the number of hypercubes of side length $\delta = \frac{\epsilon(1-\gamma)}{2\gamma}$ that cover the convex polytope of reward functions implied by the initial IRL constraints on the agent's policy and the admissibility condition on $\Delta$.

This result holds true regardless of how $\widehat{R}$ is picked in line 6 of Algorithm 1. As we will show, much tighter bounds can be obtained for specific elicitation strategies.

### Elicitation Objective Function

The elicitation method allows for any elicitation strategy to be used for choosing $\widehat{R}$ and $\widehat{\Delta}$ in each round. Good elicitation strategies are computationally tractable, lead to few elicitation rounds, and provide robustness guarantees (e.g., provide useful bounds on the number of elicitation rounds or minimize max regret). We present a centroid-based strategy with nice properties, as well as a practical two-sided slack maximization heuristic.

**Lemma 1.** Let $B_K^t$ denote the (convex) set of reward functions $R$ satisfying the constraints in $K$ before the $t$-th iteration of Algorithm 1. Let $c_t$ denote the centroid of $B_K^t$, and consider an elicitation strategy that picks $\widehat{R} = c_t$ and any corresponding admissible $\widehat{\Delta}$. Then $\widehat{\Delta}$ will either induce $\pi_T$ or the added IRL constraints will eliminate at least $\frac{1}{e}$ of the volume of $B_K^t$ (that is, $vol(B_K^{t+1}) \leq (1 - \frac{1}{e})vol(B_K^t)$).

This lemma makes use of Grünbaum's result that any halfspace containing the centroid of a convex set in $\mathbb{R}^n$ contains $\frac{1}{e}$ of its volume (1960). Since $B_K^{t+1}$ is a closed convex set that does not contain the eliminated centroid, an application of the separating hyperplane theorem gives the desired result. Since each iteration cuts off a constant fraction of the volume of the reward space, the following bound on the number of elicitation rounds applies:

**Theorem 4.** *Consider any elicitation strategy that picks the centroid of $B_K^t$ for $\widehat{R}$ in Algorithm 1. The number of elicitation rounds is bounded above by $|S| \log_b \frac{R_{max}}{\delta}$, where $b = \frac{1}{1-\frac{1}{e}}$ and $\delta = \frac{\epsilon(1-\gamma)}{2\gamma}$.*

Here $(\frac{R_{max}}{\delta})^{|S|}$ is the number of hypercubes with side length $\delta$ that can fit within the bounded space of rewards considered. This can be viewed as the size of the problem, and the bound given by Theorem 4 is logarithmic (versus linear in the convergence bound from Theorem 3).

Computing the centroid exactly is #P-hard (Rademacher 2007), but polynomial time randomized algorithms can approximate it (Bertsimas and Vempala 2004). Furthermore, Bertsimas and Vempala extend Grunbaüm's result to the case of the approximate centroid, such that with $O(|S|)$ uniform samples, any halfspace through the average of the samples will cut off a constant fraction of the volume of a convex set with arbitrarily high probability.

The following theorem offers an elicitation strategy that allows $\widehat{R}$ to be computed in polynomial time while guaranteeing elicitation convergence in rounds logarithmic in the problem size with high probability.

**Theorem 5.** *Consider any elicitation strategy that picks the average of $O(|S|)$ points sampled uniformly from $B_K^t$ for $\widehat{R}$ in Algorithm 1. With arbitrarily high probability, Algorithm 1 terminates before $|S| \log_b \frac{R_{max}}{\delta}$ rounds, where $b = \frac{1}{1-k}$ for a constant fraction $k < \frac{1}{e}$.*

Since sampling $O(|S|)$ points takes $O(|S|^4)$ steps of a random walk that takes $O(|S|^2)$ operations per step, computing $\widehat{R}$ this way is $O(|S|^6)$ (Bertsimas and Vempala 2004). One still has to find some corresponding $\widehat{\Delta}$, but this need only require solving a simple linear program (e.g., the one in Theorem 2). Nevertheless, it seems likely that an algorithm based on this elicitation strategy may not scale well in practice for large state spaces.

In thinking about more practical methods, we consider the effect of the slack in the IRL constraints. This slack corresponds to the amount of perturbation allowed in the reward function without changing the optimal policy. For any $\widehat{R} \in B_k^t$, there is an associated agent-side slack over the IRL constraints defining the agent's IRL space. For any $\widehat{\Delta}$ mapping from $\widehat{R}$, there is an associated target-side slack over the IRL constraints on the target policy. If we choose $\widehat{\Delta}$ to induce the target policy with high slack, then through a 1:1 mapping from $\widehat{R}$ to $\widehat{R}_T$, failure to induce the target policy results in a large volume of points around $\widehat{R}$ that can't be the agent's reward. If $\widehat{R}$ is sufficiently far from the boundaries of $B_K^t$ (e.g., near the centroid), then this volume of points will be within $B_K^t$. Since the convex polytope defined by the added IRL constraints cannot contain any points within this volume, choosing $\widehat{\Delta}$ to maximize the target-side slack provides a complementary method for making large cuts in the IRL space.

As an alternative approach to approximating the centroid, we can attempt to find $\widehat{R}$ such that a large volume of points around $\widehat{R}$ are contained in $B_K^t$ and thus can be eliminated via the target mapping. This suggests a *two-sided slack maximization* heuristic which finds $\widehat{R}$ and $\widehat{\Delta}$ to maximize the minimal slack over all IRL constraints on the agent's initial policy $\pi$, induced policies $\pi'$, and target policy based on $\widehat{R} + \widehat{\Delta}$. By using a single slack for the agent's reward space and the target space, we simultaneously push $\widehat{R}$ towards the center of the agent's IRL space and bound a large volume around it that will be eliminated if the mapping fails.

To formulate the heuristic as a linear program, we introduce a single variable $\beta \geq 0$ to capture this minimal slack, and introduce $\alpha(s)$ variables that capture the absolute value of $\widehat{R}$. Using $\alpha$ as a $\lambda$-weighted penalty (for some constant $\lambda > 0$) on the size of rewards picked, we have the following objective function and associated constraints:

$$\max\ \beta - \lambda \sum_s \alpha(s) \tag{7}$$

$$\begin{aligned}
((\mathbf{P}_\pi - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1}\mathbf{R})[s] &\geq \beta && \forall a, s \\
((\mathbf{P}_{\pi'} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-1}(\mathbf{R} + \mathbf{\widehat{\Delta}}))[s] &\geq \beta && \forall a, s, \pi' \\
((\mathbf{P}_{\pi_\mathbf{T}} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi_\mathbf{T}})^{-1}\mathbf{R_T})[s] &\geq \beta && \forall a, s \\
\alpha(s) &\geq R(s) && \forall s \\
\alpha(s) &\geq -R(s) && \forall s \\
\beta &\geq 0
\end{aligned}$$

We can use this max-slack heuristic to find $\widehat{R}$ and $\widehat{\Delta}$ in Algorithm 1 by solving a linear program containing the above equations and also the constraints from the constraint set $K$. In round $t$ of the elicitation process, the algorithm will generate a linear program with approximately $2t|S||A|$ constraints, which can then be solved efficiently using standard techniques. This method provides a tractable alternative to approximating the centroid and has been effective when evaluated empirically.[4]

[4] In a simulated advertising network setting with an advertiser trying to induce a publisher to design the hyperlinks of a website in a desired way, our results show an average number of elicitation rounds of around 10 for MDPs with up to 100 states. An experimental results section is omitted here due to space considerations.

## Generalizing to environment design

Having provided methods for the problem of policy teaching, we discuss generalizations of our approach to allow for other kinds of modifications to an agent's environment.

**Definition 5. Environment Design with unknown rewards.** An agent follows an optimal policy $\pi$ with respect to an MDP $M = \{S, A, R, P, \gamma\}$. An interested party observes $\pi$ and has knowledge of $M_{-R} = \{S, A, P, \gamma\}$, but does not know the agent's reward. The interested party can alter the environment via an admissible change $\Pi$, which via a perturbation function $f$ induces the agent to act with respect to $M' = f(M, \Pi) = \{S', A', R', P', \gamma'\}$ in the perturbed environment. Find an admissible $\Pi$ to induce the agent to perform some desired policy $\pi_T$ and strictly prefer this to any other policy (or show that no such $\Pi$ is available).

Assuming that the perturbation function $f$ is deterministic, that is, the environment will change in the way it is intended to change, our active indirect elicitation method can be used with little modification. At every interaction, the interested party can pick a $\widehat{R}$ with an admissible $\widehat{\Pi}$ that would induce an agent with reward $\widehat{R}$ to follow the desired policy. If the agent performs a policy $\pi' \neq \pi_T$ in the perturbed environment $M'$, we can add a set of IRL constraints based on $M'$ and $\pi'$, where $R$ is still the only variable in these constraints. Assuming the state space does not change[5], by an identical argument to that used in Theorem 3, this process is still convergent. Furthermore, assuming that the space of possible agent rewards with admissible $\Pi$ mappings to the desired policy is convex, the logarithmic bounds on the number of elicitation rounds via a centroid-based elicitation strategy continue to hold.

An important question for future work is to explore the powers and limitations of these alternate "levers" to induce desirable behaviors. This will depend partly on the admissibility conditions on alterations to the environment for particular domains. Furthermore, since perturbations to the environment are likely to have stochastic effects, one may wish to interleave learning the agent's reward with learning the effects of changes on the resulting environment. On the computational side, note that unlike linear transformations to the reward function in policy teaching, figuring out the set of environments that induce a particular policy may be quite expensive; efficient algorithms will be important.

We also intend to consider multi-agent variants, both with multiple agents acting in the environment and with multiple interested parties, where each interested party is able to modify a portion of the complete environment, and each with individual objectives for influencing the agent's decisions in some subset of the state space. Other directions for future work include a more careful analysis of incentives and loosening the assumption of myopic-rationality on the agent for general forms of environment design, as well as considerations for uncertainty over other aspects of the agent's model and behavior beyond the reward function (e.g., a partially observable state space).

[5]This is to ensure that $\pi_T$ and $S'$ are of equal dimensions, which simplifies the discussion.

## Conclusions

We study the interesting new paradigm of environment design. The problem requires indirect preference elicitation; we provide a general active, indirect elicitation framework that allows us to quickly learn an agent's preferences and converge to good environments.

## References


Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM Press.

Bertsimas, D., and Vempala, S. 2004. Solving convex programs by random walks. *J. ACM* 51(4):540–556.

Boutilier, C.; Patrascu, R.; Poupart, P.; and Schuurmans, D. 2005. Regret-based utility elicitation in constraint-based decision problems. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, 929–934.

Boutilier, C.; Sandholm, T.; and Shields, R. 2004. Eliciting bid taker non-price preferences in (combinatorial) auctions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*, 204–211.

Chajewska, U.; Koller, D.; and Ormoneit, D. 2001. Learning an agent's utility function by observing behavior. In *Proc. 18th International Conf. on Machine Learning*, 35–42. Morgan Kaufmann, San Francisco, CA.

Chajewska, U.; Koller, D.; and Parr, R. 2000. Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 363–369. AAAI Press / The MIT Press.

Gajos, K., and Weld, D. S. 2005. Preference elicitation for interface optimization. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, 173–182. New York, NY, USA: ACM.

Grunbaum, B. 1960. Partitions of mass-distributions and of convex bodies by hyperplanes. *Pacific Journal of Mathematics* 10(4):1257–1261.

Ng, A. Y., and Russell, S. 2000. Algorithms for inverse reinforcement learning. In *Proc. 17th International Conf. on Machine Learning*, 663–670. Morgan Kaufmann, San Francisco, CA.

Puterman, M. L. 1994. *Markov decision processes: Discrete stochastic dynamic programming*. New York: John Wiley & Sons.

Rademacher, L. A. 2007. Approximating the centroid is hard. In *SCG '07: Proceedings of the twenty-third annual symposium on Computational geometry*, 302–305. New York, NY, USA: ACM.

Sandholm, T., and Boutilier, C. 2006. Preference elicitation in combinatorial auctions. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. chapter 10.

Varian, H. 2003. Revealed preference. In Szenberg, M., ed., *Samuelsonian Economics and the 21st Century*. Oxford University Press.

Wang, T., and Boutilier, C. 2003. Incremental utility elicitation with the minimax regret decision criterion. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*.

Zhang, H., and Parkes, D. 2008. Value-based policy teaching with active indirect elicitation. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence (AAAI-2008)*.