Value-Based Policy Teaching with Active Indirect Elicitation

Haoqi Zhang and **David 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's utility is dependent on the actions of an agent; e.g., a teacher is interested in a student learning effectively and a firm is interested in a consumer's behavior. We consider an environment in which the interested party can provide incentives to affect the agent's actions but cannot otherwise enforce actions. In *value-based policy teaching*, we situate this within the framework of sequential decision tasks modeled by Markov Decision Processes, and seek to associate limited rewards with states that induce the agent to follow a policy that maximizes the total expected value of the interested party. We show value-based policy teaching is NP-hard and provide a mixed integer program formulation. Focusing in particular on environments in which the agent's reward is unknown to the interested party, we provide a method for *active indirect elicitation* wherein the agent's reward function is inferred from observations about its response to incentives. Experimental results suggest that we can generally find the optimal incentive provision in a small number of elicitation rounds.

Introduction

Many situations arise in which an interested party's utility depends on the actions of an agent. For example, a teacher wants a student to develop good study habits. Parents want their child to come home after school. A firm wants a consumer to make purchases. Often, the behavior desired by the interested party differs from the actual behavior of the agent. The student may be careless on homeworks, the child may go to the park and not come home, and the consumer may not buy anything.

The interested party can often provide incentives to encourage desirable behavior. A teacher can offer a student gold stars, sweets, or prizes as rewards for solving problems correctly. Parents can motivate their child to come home by allowing more TV time or providing money for a snack on the way home. A firm can provide product discounts to entice a consumer to make a purchase.

We view these incentive provision problems as problems of *policy teaching*: given an agent behaving in an environment, how can an interested party provide incentives to induce a desired behavior? We consider a setting in which the agent performs a sequence of observable actions, repeatedly and relatively frequently. The interested party has measurements of the agent's behavior over time, and can modify the environment by associating additional rewards with world states or agent actions. The agent may choose to behave differently in the modified environment, but the interested party cannot otherwise impose actions upon the agent. Later in the paper we situate the problem of policy teaching within the general problem of *environment design*.

In *value-based policy teaching*, we adopt the framework of sequential decision tasks modeled by Markov Decision Processes (MDPs), and seek to provide incentives that induce the agent to follow a policy that maximizes the total expected value of the interested party, subject to constraints on the total amount of incentives that can be provided. While the problem of providing incentives to induce a *particular* policy can be solved with a linear program (Zhang and Parkes 2008), the problem here is NP-hard. In general, the interested party may find many agent policies desirable, of which only a subset are teachable with limited incentives. For example, a teacher may find many study methods desirable but would not know ahead of time which is the most effective study method that a student can be motivated to follow given a limited number of gold stars.

Computational challenges aside, the agent's local rewards (and thus its utility function) may also be unknown to the interested party, further complicating the problem of policy teaching. A common approach to preference elicitation is to ask the agent a series of direct queries about his or her preferences, based on which bounds can be placed on the agent's utility function (Boutilier et al. 2005; Chajewska, Koller, and Parr 2000; Wang and Boutilier 2003). The direct elicitation approach has been critiqued over concerns of practicality, as certain types of queries may be too difficult for an agent to answer and the process may be error-prone (Chajewska, Koller, and Ormoneit 2001; Gajos and Weld 2005). While we share these practical concerns, we are also opposed to using direct elicitation for policy teaching because it is intrusive: given that the interested party cannot impose actions upon the agent, there is little reason to believe that the interested party can enforce participation in a costly external elicitation process.

In this paper, we address both the computational and elicitation challenges in value-based policy teaching. On the

computational side, we provide a novel mixed integer program (MIP) formulation that can be used to solve reasonably sized problem instances. For learning the agent's preferences, we provide an *active indirect elicitation* method wherein the agent's reward function is inferred from observations of the agent's policy in response to incentives. Using this method, we construct an algorithm that uses lower bounds on the value of the best teachable policy and the constraints of *inverse reinforcement learning* (Ng and Russell 2000) on induced policies to continually narrow the space of possible agent rewards until the best teachable policy found so far is guaranteed to be within some bound of the optimal solution.

Inverse reinforcement learning (IRL) considers the problem of determining a set of rewards consistent with an observed policy. But this is insufficient here, because the actual value of the underlying rewards matters in finding the right adjustment via incentives. By iteratively modifying the agent's environment and observing new behaviors we are able to make progress towards the optimal, teachable policy. We prove that the method will converge in a bounded number of rounds, and also provide a *two-sided slack maximization* heuristic that can significantly reduce the number of elicitation rounds in practice. Experimental results show that elicitation converges in very few rounds and the method scales to moderately sized instances with 20 states and 5 actions.

Related Work

Other works on preference elicitation have also taken the indirect approach of inferring preferences from observed behavior. Indeed, this is the essential idea of *revealed preference* from microeconomic theory (Varian 2003). As noted above, for MDPs, Ng and Russell (2000) introduced the problem of IRL and showed that reward functions consistent with an optimal policy can be expressed by a set of linear inequalities. Later works have extended IRL to a Bayesian framework (Chajewska, Koller, and Ormoneit 2001; Ramachandran and Amir 2007), but techniques remained passive; they are applied to observed behaviors of an agent acting in a particular environment (e.g., with respect to an unchanging MDP), and are unconcerned with generating new evidence from which to make further inferences about preferences.

To our knowledge, neither the computational problem nor the learning approach have been previously studied in the literature. In work on k-implementation, Monderer and Tennenholtz (2003) studied a problem in which an interested party assigns monetary rewards to influence the actions of agents in games. But the setting there is in many ways different, as the focus is on single-shot games (no sequential decisions) and the game is assumed known (no elicitation problem). Furthermore, unlike our work here, the interested party in k-implementation has the power to assign unlimited rewards to states, and relies on this credibility to implement desirable outcomes.

The problem of policy teaching is closely related to principal-agency problems studied in economics, where a principal (e.g., firm) provides incentives to align the interest of the agent (e.g., employee) with that of the principal (Bolton and Dewatripont 2005; Laffront and Martimort 2001). But the focus in principal-agency theory is rather different, dealing mainly with "*moral hazard*" problems of hidden actions, and situated in the large part in simpler environments and in models for which the preferences of the agent are known.[1]

Problem Definition

We model an agent performing a sequential decision task with an infinite horizon MDP $M = \{S, A, R, P, \gamma\}$, where S is the set of states, A is the set of possible actions, $R : S \to \mathbb{R}$ is the reward function, $P : S \times A \times S \to [0, 1]$ is the transition function, and γ is the discount factor from $(0, 1)$. We assume finite state and action spaces, and consider agent rewards bounded in absolute value by R_{max}.

Given an MDP, the agent's goal is to maximize the expected sum of discounted rewards. We consider the agent's decisions as a stationary policy π, such that $\pi(s)$ is the action the agent executes in state s. Given a policy π, the value function $V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P(s, \pi(s), s') V^\pi(s')$ is uniquely defined and captures the expected sum of discounted rewards under π. Similarly, the Q function captures the value of taking an action a followed by policy π 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, an optimal policy π^* chooses actions that maximize the Q function in every state, such that $\pi^*(s) \in \operatorname{argmax}_{a \in A} Q^{\pi^*}(s, a)$ (Puterman 1994). We assume the agent is capable of solving the planning problem and performs with respect to π^*.

Given an agent following a policy π, we represent the interested party's reward and value functions by G and V_G^π respectively, such that $V_G^\pi(s) = G(s) + \gamma \sum_{s' \in S} P(s, \pi(s), s') V_G^\pi(s')$. The interested party can provide incentives to the agent through an incentive function Δ. Given a start state $start$,[2] we define the notion of admissibility:

Definition 1. An incentive function $\Delta : S \to \mathbb{R}$ is *admissible* with respect to π_T if it satisfies the following constraints:

$$V_\Delta^{\pi_T}(s) = \Delta(s) + \gamma P_{s,\pi_T(s)} V_\Delta^{\pi_T}, \forall s \qquad \textit{Incentive value.}$$

$$V_\Delta^{\pi_T}(start) \leq D_{max} \qquad \textit{Limited spending.}$$

$$0 \leq \Delta(s) \leq \Delta_{max}, \forall s \qquad \textit{No punishments.}$$

This notion of admissibility limits the expected incentive provision to D_{max} when the agent performs π_T from the start state. We assume the agent's reward is state-wise quasi-linear, such that after providing incentives the agent plans with respect to $R' = R + \Delta$. Here we have also assumed that the agent is *myopically rational*, in that given Δ, the agent plans with respect to $R + \Delta$ and does not reason about incentive provisions in future interactions with the interested party.

[1] See Feldman et al. (2005) and Babaioff et al. (2006) for recent work in computer science related to principal-agent problems on networks.

[2] The use of a single start state is without loss of generality, since it can be a dummy state whose transitions represent a distribution over possible start states.

We aim to find an admissible Δ that induces a policy that maximizes the value of the interested party:

Definition 2. Given a policy π and $M_{-R} = \{S, A, P, \gamma\}$, let $R \in IRL^{\pi}$ denote the space of all reward functions R for which π is optimal for the MDP $M = \{S, A, R, P, \gamma\}$.

Definition 3. Let $OPT(R, D_{max})$ denote the set of pairs of incentive functions and teachable policies, such that $(\Delta, \pi') \in OPT(R, D_{max})$ if and only if Δ is admissible with respect to π' given D_{max}, and $(R + \Delta) \in IRL^{\pi'}$.

Definition 4. *Value-based policy teaching with known rewards.* Given an agent MDP $M = \{S, A, R, P, \gamma\}$, incentive limit D_{max}, and the interested party's reward function G, find $(\Delta, \pi') \in \operatorname{argmax}_{(\widehat{\Delta},\widehat{\pi}) \in OPT(R, D_{max})} V_G^{\widehat{\pi}}(start)$.

Theorem 1. *The value-based policy teaching problem with known rewards is NP-hard.*

Proof. We perform a reduction from KNAPSACK (Garey and Johnson 1979). Given n items, we denote item i's value by v_i and its weight by c_i. With a maximum capacity C, a solution to the knapsack problem finds the set of items to take that maximizes total value and satisfies the capacity constraint. For our reduction, we construct an agent MDP with $2n + 2$ states. The agent has a `leave_it` action a_0 and a `take_it` action a_1. Starting from state s_0, the agent transitions from state s_{i-1}^k to s_i^j on action a_j for an arbitrary k, where the sequence of states visited represents the agent's decisions to take or leave each item. Once all decisions are made (when state s_n^k is visited for an arbitrary k), the agent transitions to a terminal state s_t.

The agent carries the weight of the items, such that $R(s_i^1) = -c_i\gamma^{-i}$. The interested party receives the value of the items, such that $G(s_i^1) = v_i\gamma^{-i}$. Rewards are 0 in all other states. Given an agent policy π, let T denote the set of items the agent takes when following π from s_0. It follows that $V^{\pi}(s_0) = -\sum_{i \in T} c_i$ is the total weight and $V_G^{\pi}(s_0) = \sum_{i \in T} v_i$ is the total value of carried items. Under reward R, the agent does not take any items; the interested party provides positive incentive Δ to induce the agent to carry items that maximize $V_G(s_0)$ while satisfying the capacity constraint $V_{\Delta}(s_0) \leq C$. Since $\Delta(s_i^1) = c_i\gamma^{-i}$ is sufficient for item i to be added to the knapsack and contribute weight c_i to $V_{\Delta}(s_0)$, the Δ that induces the policy with the highest $V_G(s_0)$ solves the knapsack problem. □

Note that the problem in Definition 4 does not explicitly factor in the cost of the provided incentives into the objective. This is in some sense without loss of generality, because an objective that maximizes expected reward net of cost still leads to a NP-hard problem that can be solved with the mixed integer programming approach we will present.[3,4]

[3]To incorporate cost, we rewrite the interested party's reward as $G' = G - \Delta$, such that maximizing the value with respect to G' maximizes the expected payoff. Note that here Δ (and thus G') is a variable. To establish NP-hardness, we use the same construction as in Theorem 1, but let $G(s_i^1) = (v_i + c_i)\gamma^{-i}$ and $G'(s_i^1) = (v_i + c_i)\gamma^{-i} - \Delta(s_i^1)$. Since $\Delta(s_i^1) = c_i\gamma^{-i}$ is sufficient for item i to be added to the knapsack, the maximizing $V_{G'}(s_0)$ maximizes the value of the knapsack.

[4]The complexity of the problem with an objective that maximizes expected payoff but with no limit on incentive provision is an open problem.

MIP Formulation

The computational difficulty of value-based policy teaching stems from the problem's indirectness: the interested party must provide limited incentives to *induce an agent policy* that maximizes the value *of the interested party*. While both the admissibility conditions and the interested party's value function are defined with respect to the agent's induced policy π', this policy is not known ahead of time but instead is a variable in the optimization problem. To capture the agent's decisions explicitly, we introduce binary variables X_{sa} to represent the agent's optimal policy π' with respect to $R + \Delta$, such that $X_{sa} = 1$ if and only if $\pi'(s) = a$. The following constraints capture the interested party's Q function Q_G and value function V_G with respect to π':

$$Q_G(s,a) = G(s) + \gamma P_{s,a} V_G \qquad \forall s, a \qquad (1)$$

$$V_G(s) = \sum\nolimits_a Q_G(s,a) X_{sa} \qquad \forall s \qquad (2)$$

Here $V_G(s) = Q_G(s,a)$ if and only if $\pi'(s) = a$. Constraint 2 is nonlinear, but we can rewrite it as a pair of linear constraints using the big-M method:

$$V_G(s) \geq -M_{gv}(1 - X_{sa}) + Q_G(s,a) \qquad \forall s, a \qquad (3)$$

$$V_G(s) \leq M_{gv}(1 - X_{sa}) + Q_G(s,a) \qquad \forall s, a \qquad (4)$$

Here M_{gv} is a large constant, which will be made tight. When $X_{sa} = 1$, $V_G(s) \geq Q_G(s,a)$ and $V_G(s) \leq Q_G(s,a)$ imply $V_G(s) = Q_G(s,a)$. When $X_{sa} = 0$, both constraints are trivially satisfied by the large constant. We apply the same technique for the value function corresponding to the agent's problem and that corresponding to the admissibility requirement.

Theorem 2. *The following mixed integer program solves the value-based policy teaching problem with known rewards:*

$$\max_{\Delta, V, Q, V_G, Q_G, V_{\Delta}, Q_{\Delta}, X} V_G(start) \qquad (5)$$

subject to:

$$Q(s,a) = R(s) + \Delta(s) + \gamma P_{s,a} V \qquad \forall s, a \qquad (6)$$

$$V(s) \geq Q(s,a) \qquad \forall s, a \qquad (7)$$

$$V(s) \leq M_v(1 - X_{sa}) + Q(s,a) \qquad \forall s, a \qquad (8)$$

$$Q_G(s,a) = G(s) + \gamma P_{s,a} V_G \qquad \forall s, a \qquad (9)$$

$$V_G(s) \geq -M_{gv}(1 - X_{sa}) + Q_G(s,a) \qquad \forall s, a \qquad (10)$$

$$V_G(s) \leq M_{gv}(1 - X_{sa}) + Q_G(s,a) \qquad \forall s, a \qquad (11)$$

$$Q_{\Delta}(s,a) = \Delta(s) + \gamma P_{s,a} V_{\Delta} \qquad \forall s, a \qquad (12)$$

$$V_{\Delta}(s) \geq -M_{\Delta}(1 - X_{sa}) + Q_{\Delta}(s,a) \qquad \forall s, a \qquad (13)$$

$$V_{\Delta}(s) \leq M_{\Delta}(1 - X_{sa}) + Q_{\Delta}(s,a) \qquad \forall s, a \qquad (14)$$

$$V_{\Delta}(start) \leq D_{max} \qquad (15)$$

$$0 \leq \Delta(s) \leq \Delta_{max} \qquad \forall s \qquad (16)$$

$$\sum\nolimits_a X_{sa} = 1 \qquad \forall s \qquad (17)$$

$$X_{sa} \in \{0, 1\} \qquad \forall s, a \qquad (18)$$

where constants $M_v = \overline{M_v} - \underline{M_v}$ *and* $M_{gv} = \overline{M_{gv}} - \underline{M_{gv}}$ *are set such that* $\overline{M_v} = (\Delta_{max} + \max_s R(s))/(1-\gamma)$, $\underline{M_v} = \min_s R(s)/(1-\gamma)$, $\overline{M_{gv}} = \max_s G(s)/(1-\gamma)$, *and* $\underline{M_{gv}} = \min_s G(s)/(1-\gamma)$. $M_\Delta = \Delta_{max}/(1-\gamma)$.

Constraint 6 defines the agent's Q functions in terms of R and Δ. Constraints 7 and 8 ensure that the agent takes the action with the highest Q value in each state. To see this, consider the two possible values for X_{sa}. If $X_{sa} = 1$, $V(s) = Q(s,a)$. By Constraint 7, $Q(s,a) = \max_i Q(s,i)$. If $X_{sa} = 0$, the constraints are satisfied because $M_v \geq \max V(s) - Q(s,a)$.[5] Constraints 9, 10, and 11 capture the interested party's value for the induced policy. Similarly, constraints 12–16 capture the admissibility conditions. Constraints 17 and 18 ensure that exactly one action is chosen for each state. The objective maximizes the interested party's value from the start state. All big-M constants have been set tightly to ensure a valid but strong formulation.

In practice, we may wish to avoid scenarios where multiple optimal policies exist (i.e., there are ties) and the agent may choose a policy other than the one that maximizes the value of the interested party. To ensure that the desired policy is uniquely optimal, we can also define a slight variant with a strictness condition on the induced policy by adding the following constraint to the mixed integer program:

$$V(s) - Q(s,a) + \epsilon X_{sa} \geq \epsilon \quad \forall s, a \tag{19}$$

where $\epsilon > 0$ is a small constant that represents the *minimal slack* between the Q value of the induced optimal action ($X_{sa} = 1$) and any other actions. We can also define ϵ-strict equivalents for IRL^π and OPT, such that $R \in IRL^\pi_\epsilon$ denotes the space of rewards that strictly induce π and $OPT_\epsilon(R, D_{max})$ denotes the set of pairs of incentive functions and strictly teachable policies.

Active Indirect Elicitation

Generally, the interested party will not know the agent's reward function. Here we make use of the notion of strictness and require the interested party to find the optimal ϵ-strict incentives.

Definition 5. *Value-based policy teaching with unknown agent reward.* An agent follows an optimal policy π with respect to an MDP $M = \{S, A, R, P, \gamma\}$. An interested party with reward function G observes $M_{-R} = \{S, A, P, \gamma\}$ and π but not R. Given incentive limit D_{max} and $\epsilon > 0$, set Δ and observe agent policy π' such that $(\Delta, \pi') \in OPT(R, D_{max})$ and $V_G^{\pi'}(start) \geq max_{(\widehat{\Delta},\widehat{\pi}) \in OPT_\epsilon(R, D_{max})} V_G^{\widehat{\pi}}(start)$.

Note that the value to the interested party is determined by the observed agent policy π'. In this definition we are allowing for non-strict incentive provisions under which the observed agent policy is of greater value than the policy corresponding to the optimal ϵ-strict incentive provision.

[5]Since $\overline{M_v}$ is the sum of discounted rewards for staying in the state with the highest possible reward and $\underline{M_v}$ is the sum of discounted rewards for staying in the state with the lowest possible reward, it must be that $\overline{M_v} \geq \max V(s)$ and $\underline{M_v} \leq \min Q(s,a)$. This implies that $M_v \geq \max V(s) - Q(s,a)$.

To begin, we can use the following theorem to classify all reward functions consistent with the agent's policy π:

Theorem 3. *(Ng and Russell 2000) Given a policy* π *and* $M_{-R} = \{S, A, P, \gamma\}$, $R \in IRL^\pi$ *satisfies:*

$$(\mathbf{P}_\pi - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1}\mathbf{R} \succeq \mathbf{0} \quad \forall a \in A \tag{20}$$

While Theorem 3 finds the space of reward functions consistent with the agent's observed behavior, the constraints do not locate the agent's actual reward within this space. To find the optimal incentive provision for the agent's true reward, it is necessary to narrow down this "IRL reward space."

The method begins by making a guess $\widehat{R}$ at the agent's reward R by choosing any point within the IRL space of the agent that has an associated admissible $\widehat{\Delta}$ such that $\widehat{R} + \widehat{\Delta}$ strictly induces a policy $\widehat{\pi}_T$ with higher value to the interested party than the agent's current policy. If our guess is correct, we would expect providing the agent with $\widehat{\Delta}$ to strictly induce policy $\widehat{\pi}_T$. If instead the agent performs a policy $\pi' \neq \widehat{\pi}_T$, we know that $\widehat{R}$ must not be the agent's true reward R. Furthermore, we also know that $R + \widehat{\Delta}$ induces π', providing additional information which may eliminate other points in the space of agent rewards. We obtain an IRL constraint on $R + \widehat{\Delta}$ such that $(R + \widehat{\Delta}) \in IRL^{\pi'}$:

$$(\mathbf{P}_{\pi'} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-1}(\mathbf{R} + \widehat{\mathbf{\Delta}}) \succeq \mathbf{0} \quad \forall a \in A \tag{21}$$

We can repeat the process of guessing a reward in the agent's IRL space, providing incentives based on the hypothesized reward, observing the induced policy, and adding new constraints if the agent does not behave as expected. *But, what if the agent does behave as expected?* Without contrary evidence, adding an IRL constraint $R + \widehat{\Delta} \in IRL^{\widehat{\pi}_T}$ does not remove $\widehat{R}$ from the agent's IRL space. While $\widehat{\Delta}$ is the optimal incentive provision for $\widehat{R}$, the optimal incentive provision from the agent's true reward may still induce a policy with a higher value.

We handle this issue by also keeping track of the most effective incentives provided so far. We initialize $V_G^{max} = V_G^\pi(start)$ for initial agent policy π. Given an induced policy π' with respect to $R + \widehat{\Delta}$, we calculate $V_G^{\pi'}$. If $V_G^{\pi'}(start) > V_G^{max}$, we update $\Delta_{best} = \widehat{\Delta}$ and $V_G^{max} = V_G^{\pi'}(start)$. In choosing $\widehat{R}$, we consider only rewards that have a strict admissible mapping to a policy that would induce $V_G(start) > V_G^{max}$. We denote the space of such rewards as $R \in R_{>V_G^{max}}$, which corresponds to satisfying constraints 6 through 19 (where R will be a variable in these constraints) and also the following constraint:

$$V_G(start) \geq V_G^{max} + \kappa \tag{22}$$

for some constant $\kappa > 0$. In each elicitation round, we find some $\widehat{R}$ that satisfies IRL constraints and is in the space $R_{>V_G^{max}}$ and provide the agent with corresponding incentive $\widehat{\Delta}$. Based on the agent's response, $\widehat{R}$ is guaranteed to be eliminated either by additional IRL constraints or by an updated V_G^{max}. If no $\widehat{R}$ satisfies IRL constraints and is in the space $R_{>V_G^{max}}$, we know there are no admissible incentives

Algorithm 1 Value-based active indirect elicitation

Require: agent policy π, interested party reward G
1: variables R, Δ; constraint set $K = \emptyset$
2: $V_G^{max} = V_G^{\pi}(start)$, $\Delta_{best} = 0$
3: Add $R \in IRL^{\pi}$, $R \in R_{>V_G^{max}}$ to K
4: **loop**
5: Find $\widehat{\Delta}$, $\widehat{R}$ satisfying all constraints in K
6: **if** no such values exist **then**
7: return Δ_{best}
8: **else**
9: Provide agent with incentive $\widehat{\Delta}$
10: Observe π' with respect to $R' = R^{true} + \widehat{\Delta}$.
11: **if** $V_G^{\pi'}(start) > V_G^{max}$ **then**
12: $V_G^{max} = V_G^{\pi'}(start)$, $\Delta_{best} = \widehat{\Delta}$.
13: Modify $R \in R_{>V_G^{max}}$ in K
14: Add $(R + \widehat{\Delta}) \in IRL^{\pi'}$ to K

from any possible agent rewards that can induce a better policy than that found so far and can end the elicitation process.

Algorithm 1 gives our elicitation method. In describing the algorithm, the set of constraints in some round is denoted by K and an instantiation of a variable R is denoted by $\widehat{R}$.

Theorem 4. *Given $\epsilon > 0$, D_{max}, and $\kappa > 0$, Algorithm 1 terminates in a finite number of steps with an admissible Δ that induces the agent to follow a policy π' with $V_G^{\pi'}(start) \geq max_{(\widehat{\Delta},\widehat{\pi}) \in OPT_{\epsilon}(R,D_{max})} V_G^{\widehat{\pi}}(start) - \kappa$.*

Proof sketch. Every iteration of Algorithm 1 finds $\widehat{R}$ and $\widehat{\Delta}$ that strictly induce a policy $\widehat{\pi}_T$ with minimal slack at least ϵ. If the agent performs a policy $\pi' \neq \widehat{\pi}_T$, the added IRL constraint $R+\widehat{\Delta} \in IRL^{\pi'}$ ensures that $\widehat{R}$ and all points within an open hypercube with side length $2\delta = \epsilon(1-\gamma)/\gamma$ centered at $\widehat{R}$ are not the agent's reward function. By a pigeonhole argument, only a finite number of $\widehat{R}$ need to be eliminated in this manner in order to cover the space of possible agent rewards. Alternatively, if $\pi' = \widehat{\pi}_T$, V_G^{max} increases by κ. Since V_G^{max} is bounded above by the value of the interested party's optimal policy, V_G^{max} can only increase a finite number of times. Since Algorithm 1 will terminate when there are no potential agent rewards that can achieve value of at least $V_G^{max} + \kappa$, we have the desired result. □

Elicitation Objective Function

The elicitation method allows for any strategy to be used for choosing some $\widehat{R}$ and $\widehat{\Delta}$ that satisfy constraints K. Desirable elicitation strategies have objective functions that are computationally tractable, find good solutions quickly, and lead to few elicitation rounds. From the convergence proof, we have seen that the size of the minimal slack around $\widehat{R}+\widehat{\Delta}$ places a bound on the volume of points around an eliminated $\widehat{R}$ that are not the agent's reward. Furthermore, if a large volume of these points lie within the agent's current IRL space (given by the intersection of all added IRL constraints up to this iteration), we can significantly narrow the space.

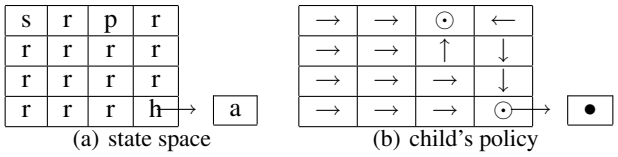

Figure 1: Child walking home domain

One heuristic approach is then to perform a *two-sided slack maximization*: find $\widehat{R}$ with a large volume of points around it that are both within the agent's IRL space and can be eliminated through the target mapping. We do this in two steps. First, we pick a reward profile that maximizes the minimal slack β across all slack on the agent's initial policy π and all induced policies π' using the following objective and associated constraints (and all existing constraints K):

$$\max_{\beta,\alpha,R,\Delta,V,Q,V_G,Q_G,V_\Delta,Q_\Delta,X} \beta - \lambda \sum_s \alpha(s) \qquad (23)$$

subject to:

$$((\mathbf{P}_\pi - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1}\mathbf{R})[s] \geq \beta \qquad \forall s, a$$
$$((\mathbf{P}_{\pi'} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-1}(\mathbf{R} + \widehat{\mathbf{\Delta}}))[s] \geq \beta \qquad \forall s, a, \pi'$$
$$\alpha(s) \geq R(s) \qquad \forall s$$
$$\alpha(s) \geq -R(s) \qquad \forall s$$
$$\beta \geq 0$$

constraints (6) – (22)

Here, $\lambda \geq 0$ is a weighted penalty term on the size of rewards which allows us to express a preference for simpler rewards and prevent the objective from picking large reward guesses for the sake of increasing the slack β.

Based on $\widehat{R}$ found using the above objective, we solve the MIP formulation from Theorem 2 with the additional strictness condition to determine the maximal $\widehat{V}_G$ that can be reached from $\widehat{R}$. We can then solve the following mixed integer program to find an admissible Δ that most strictly induces a policy with value $\widehat{V}_G$ by maximizing the minimal slack β across all slack on the target policy:

$$\max_{\beta,\Delta,V,Q,V_G,Q_G,V_\Delta,Q_\Delta,X} \beta \qquad (24)$$

subject to:

$$V(s) - Q(s,a) + M_v X_{sa} \geq \beta \qquad \forall s, a$$
$$V_G(start) \geq \widehat{V}_G$$

constraints (6) – (19)

Experiments

The goal of our experiments is to evaluate the scalability of the mixed integer program and the effectiveness of the elicitation method with various heuristics. The algorithm is implemented in JAVA, using JOPT[6] as an interface to CPLEX

[6] http://econcs.eecs.harvard.edu/jopt

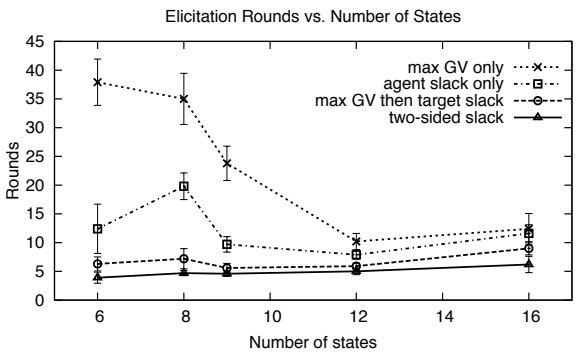

Figure 2: Elicitation rounds for various heuristics.

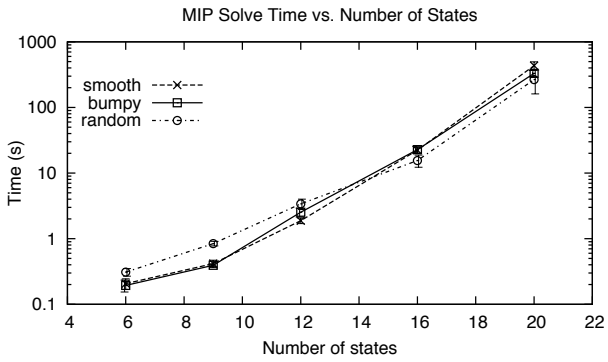

Figure 3: MIP solve time for increasing problem sizes.

(version 10.1), which served as our back-end MIP solver. Experiments were conducted on a local machine with a Pentium IV 2.4Ghz processor and 2GB of RAM.

We simulate a domain in which a child is walking home from school; see Figure 1(a). Starting at school ('s'), the child may walk in any compass direction within bounds or choose to stay put. Figure 1(b) shows a child whose policy is to go to the park and then stay there. While the child may have positive rewards for getting home (at which point he transitions into an absorbing state), he also enjoys the park, dislikes the road, and discounts future rewards. The parents are willing to provide limited incentives to induce the child to come home, preferably without a stop at the park.

We model the problem as an MDP and randomly generate instances on which to perform our experiments. We consider three separate instance generators, corresponding to *smooth* (same reward for all road states), *bumpy* (random reward over road states), and completely *random* (uniformly distributed reward for all states). For smooth and bumpy instances, the child's reward is sampled uniformly at random from [0.5, 1.5] for the park state and from [1, 2] for the home state, whereas the parent's reward is sampled uniformly at random from [-1.5, -0.5] for the park state and from [2, 4] for the home state. The incentive limit D_{max} of the interested party is set such that second-best policies (e.g., visit the park and then come home) are teachable but first-best policies (e.g., come home without visiting the park) are rarely teachable. The discount factor γ, minimal slack ϵ, and value tolerance κ are fixed at 0.7, 0.01, and 0.01, respectively.

To evaluate the elicitation algorithm, we consider four heuristics that correspond to first choosing $\widehat{R}$ to either maximize the agent-side slack (using MIP 23) or to maximize V_G across all rewards in the agent's IRL space, and then choosing whether to maximize the target-side slack (using MIP 24) after finding the maximal V_G with respect to $\widehat{R}$ or to ignore this step. We continue the elicitation process until the process converges (i.e., when the highest V_G with respect to the unknown agent reward is found *and* proved to be the best possible solution), or if the number of rounds reaches 50. All results are averaged over 10 instances.

Figure 2 shows the elicitation results. The two-sided max slack heuristic performed best, averaging less than 7 rounds for all problem sizes. Maximizing V_G and then the target slack was also effective, both via the large volumes of points eliminated through the target slack and because elicitation converges as soon as the best mapping is found. Maximizing V_G alone performed poorly on smaller instances, mostly due to its inability to induce a good policy and eliminate large volumes early (which it was able to do on the larger instances). All heuristics that we considered averaged under 15 rounds for even the larger problem instances, demonstrating the broad effectiveness of the elicitation method.

Figure 3 shows the MIP solve times on a logarithmic scale. Problems with 16 states were solved in approximately 20 seconds and problems with 20 states in 4–7 minutes. This growth in solve time for the MIP formulations appears to be exponential in the number of states, but this is perhaps unsurprising when we consider that the set of policies increases exponentially in the number of states, e.g. with up to 5^{20} target policies to naively check by straightforward enumeration. For the different types of instances, we find that the running time correlates with the ease to find the optimal target policy; e.g., target policies of random instances tend to find states near the start state with a large reward for the interested party.

For unknown agent rewards we can expect to achieve improved computational performance with tighter bounds on R_{max} because the "big-M" constraints in the MIP formulation depend heavily on R_{max}. (In the current experiments we set R_{max} to twice of the largest reward possible in the domain.) Nevertheless, an exploration of this is left to future work, along with addressing scalability to larger instances for example through identifying tighter alternative MIP formulations or through decompositions that identify useful problem structure.

Discussion: Environment Design

We view this work as the first step in a broader agenda of *environment design*. In environment design, an interested party (or multiple interested parties) act to modify an environment to induce desirable behaviors of one or more agents. The basic tenet of environment design is that it is *indirect*: rather than collect preference information and then enforce an outcome as in *mechanism design* (Jackson 2003), in envi-

ronment design one observes agent behaviors and then modulates an environment (likely at a cost) through constraints, incentives and affordances to promote useful behaviors.

To be a little more specific, we can list a few interesting research directions for future work:

- **Multiple interested parties.** Each interested party is able to modify a portion of the complete environment, and has its own objectives for influencing one or more agents' decisions. For example, what if different stakeholders, e.g. representing profit centers within an organization, have conflicting goals in terms of promoting behaviors of the users of a content network?
- **Multi-agent policy teaching.** Just as an interested party may wish to influence the behavior of a particular agent, it may also wish to influence the joint behavior of a group of agents. One possible approach is to consider the joint agent policy as meeting some equilibrium condition, and find a space of rewards consistent with the joint behavior. One can then perform elicitation to narrow this space to find an incentive provision that induces an equilibrium behavior desired by the interested party.
- **Learning agents.** If we relax the assumption that agents are planners, we enter the problem space of *reinforcement learning* (RL), where agents are adjusting towards an optimal local policy. One goal considered in the RL literature is reward shaping, where one attempts to speed up the learning of an agent by providing additional rewards in the process (Ng, Harada, and Russell 1999). Could this be adapted to settings with unknown agent rewards?
- **A Bayesian framework.** A Bayesian framework would allow us to more precisely represent uncertainty about an agent's preferences. Bayesian extensions for IRL have been considered in the literature (Chajewska, Koller, and Ormoneit 2001; Ramachandran and Amir 2007), and may be applicable to the environment design setting.
- **Alternative design levers.** Incentive provision is one of many possible ways to modify an agent's environment. One may change the physical or virtual landscape, e.g., by building a door where a wall existed or designing hyperlinks. Such changes alter the agent's model of the world, leading to different behaviors. How the general paradigm of policy teaching can be extended to include alternate design levers presents an interesting problem for future research.

Conclusions

Problems of providing incentives to induce desirable behavior arise in education, commerce, and multi-agent systems. In this paper, we introduced the problem of value-based policy teaching and solved this via a novel MIP formulation. When the agent's reward is unknown, we propose an active indirect elicitation method that converges to the optimal incentive provision in a small number of elicitation rounds. Directions for future work include extending the policy teaching framework, improving the scalability of our formulation, and building a general framework to capture the relations among environment, preferences, and behavior. There is also the question of allowing for strategic agents, which is not handled in the current work.

Acknowledgments

We thank Jerry Green, Avi Pfeffer, and Michael Mitzenmacher for reading early drafts of the work and providing helpful comments.

References

Babaioff, M.; Feldman, M.; and Nisan, N. 2006. Combinatorial agency. In *Proc. 7th ACM Conference on Electronic Commerce (EC'06)*, 18–28.

Bolton, P., and Dewatripont, M. 2005. *Contract Theory*. MIT Press.

Boutilier, C.; Patrascu, R.; Poupart, P.; and Schuurmans, D. 2005. Regret-based utility elicitation in constraint-based decision problems. In *Proc. 19th International Joint Conf. on Artificial Intelligence (IJCAI-05)*, 929–934.

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.

Chajewska, U.; Koller, D.; and Parr, R. 2000. Making rational decisions using adaptive utility elicitation. In *AAAI*, 363–369.

Feldman, M.; Chuang, J.; Stoica, I.; and Shenker, S. 2005. Hidden-action in multi-hop routing. In *Proc. 6th ACM Conference on Electronic Commerce (EC'05)*.

Gajos, K., and Weld, D. S. 2005. Preference elicitation for interface optimization. In *ACM UIST '05*, 173–182. ACM.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Jackson, M. O. 2003. Mechanism theory. In Derigs, U., ed., *The Encyclopedia of Life Support Systems*. EOLSS Publishers.

Laffront, J.-J., and Martimort, D. 2001. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press.

Monderer, D., and Tennenholtz, M. 2003. k-implementation. In *EC '03: Proc. 4th ACM conference on Electronic Commerce*, 19–28.

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

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations: theory and application to reward shaping. In *Proc. 16th International Conf. on Machine Learning*, 278–287.

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

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *IJCAI '07: Proc. 20th International joint conference on artificial intelligence*.

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 *Proc. 18th International Joint Conf. on Artificial Intelligence*.

Zhang, H., and Parkes, D. 2008. Enabling environment design via active indirect elicitation. Technical report, Harvard University. http://eecs.harvard.edu/~hq/papers/envdesign-techreport.pdf.