Policy Teaching Through Reward Function Learning*

Haoqi Zhang
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
hq@eecs.harvard.edu

David C. Parkes
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
parkes@eecs.harvard.edu

Yiling Chen
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
yiling@eecs.harvard.edu

ABSTRACT

Policy teaching considers a Markov Decision Process setting in which an interested party aims to influence an agent's decisions by providing limited incentives. In this paper, we consider the specific objective of inducing a pre-specified desired policy. We examine both the case in which the agent's reward function is known and unknown to the interested party, presenting a linear program for the former case and formulating an active, indirect elicitation method for the latter. We provide conditions for logarithmic convergence, and present a polynomial time algorithm that ensures logarithmic convergence with arbitrarily high probability. We also offer practical elicitation heuristics that can be formulated as linear programs, and demonstrate their effectiveness on a policy teaching problem in a simulated ad network setting. We extend our methods to handle partial observations and partial target policies, and provide a game-theoretic interpretation of our methods for handling strategic agents.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Economics, Theory

1. INTRODUCTION

Many situations arise in which an interested party wishes for an agent to follow desired behaviors. A retailer such as Amazon wants customers to make frequent purchases and write product reviews. Web 2.0 sites such as Facebook and YouTube want users to contribute content, show interest in advertisements, and generally spend time on the site. ad networks such as Google AdSense, Yahoo! Publisher Network, and Microsoft pubCenter want publishers to design their web sites in ways that facilitate effective advertising.

In such situations, an interested party can often make limited changes to an agent's environment to induce desired behaviors. A retailer can provide discounts on products and recognize top reviewers. A Web 2.0 site can tweak its interface to ease the content uploading process, and can reward users for performing tasks by offering money, scrips, or access to premium content. An ad service provider can offer a larger share of the advertising revenue or waive sign-up fees to entice a publisher to choose particular web layouts.

We view these problems as examples of *environment design* [20]. In environment design, an interested party aims to influence agent decisions via limited changes to the agents' environment. Private information is *indirectly* inferred from observing agents over repeated interactions, and the goal is ultimately to *elicit desired actions* from the agents. This is in contrast to mechanism design problems studied in microeconomics and computer science, where an interested party designs rules of a game to directly elicit private information from agents so that the center can make and enforce correct system-wide decisions [11]. Whereas mechanism design centers around the transmission of private information, environment design is grounded in the agents' physical actions.

In this paper, we study the particular environment design problem of *policy teaching* [21]. Policy teaching considers a Markov Decision Process (MDP) setting in which an interested party can associate limited rewards with world states to affect an agent's policy. The interested party can observe the agent's decisions in response to provided incentives, but generally does not know the agent's reward function. The interested party can interact multiple times with the agent, but cannot directly impose actions on the agent. The goal of the interested party is to induce the agent to follow a desired behavior or policy quickly and at a low cost.

Zhang and Parkes [21] previously studied *value-based* policy teaching, where the objective is to induce a policy of maximal value for the interested party across those that can be achieved with limited incentives. In this paper, we consider policy teaching in which the goal is to induce a fixed, *pre-specified desired policy*. A practical advantage of this framework is that it may be easier for the interested party to specify a desired policy than to specify a value function. This is particularly useful in many web settings where a firm knows which agent actions it desires but may have trouble estimating the value of such actions.

*An early version of this paper appeared at the AAAI 4th Multidisciplinary Workshop on Advances in Preference Handling, Chicago IL, July 2008.

EC'09, July 6–10, 2009, Stanford, California, USA.

The simpler, target-policy objective also leads to computational and convergence speed advantages. In the case that the interested party knows the agent's reward function, Zhang and Parkes [21] showed that value-based policy teaching is NP-hard. Here we show that the policy teaching problem to induce a pre-specified desired policy can be formulated as a linear program (LP). In the more likely case where the agent's reward function is unknown, we apply general environment design results [20] to guarantee elicitation convergence after logarithmic rounds. Furthermore, we apply results from sampling in convex spaces [4] to arrive at a polynomial time algorithm that ensures logarithmic elicitation convergence with arbitrarily high probability, whereas neither guarantees are available in value-based policy teaching [21]. We also offer practical elicitation heuristics, based on a two-sided max slack heuristic [21], that can be formulated as linear programs. We consider as an example a policy teaching problem in the *ad network* setting, in which an ad network can provide limited incentives to induce a publisher to choose a hyperlink design that is conducive to advertising. We implement our methods and elicitation heuristics, and provide preliminary results in a simulated ad network to demonstrate effectiveness.

In some situations, an interested party may only observe the agent's actions in a subset of the states, either because some states are private to the agent or because the agent only visits a subset of the states when following its policy. Also, an interested party may only wish to influence the agent's policy in a subset of the states. In both cases, the interested party cannot simply ignore the other states because the agent's policy depends on its rewards and actions in all the states. We extend our methods to handle policy teaching with *partial observability and partial goals*, and prove that the elicitation process remains convergent.

For the most part we assume that the agent is *myopic*, in the sense that the agent plans with respect to the current environment and does not reason about future incentives from the interested party. This assumption seems reasonable on the web, where a user of a web site tends to do what is best for himself under the current environment and is generally unconcerned about how his actions may influence future changes to the web site. Despite this assumption, we are also interested in handling both myopic *and* strategic, forward-looking agents. In this direction, we show that under certain conditions, a strategic agent responding to the active, indirect elicitation method will also be incentivized to follow the fixed desired policy as a myopic best-response.

1.1 Related work

The closest analog to policy teaching is the principal-agent problem studied in economics [5, 13], in which a principal can provide incentives in the form of contracts to align the interest of an agent with that of the principal.[1] In the MDP setting, Zhang and Zenios [22] studied a dynamic principal-agent problem in which the interested party can design the reward structure of an agent MDP but cannot observe the agent's state. Our basic setup assumes that states and actions are observable, and thus does not include moral-hazard problems. On the other hand, we have additional challenges because the agent has intrinsic rewards that are unknown to the interested party and that can only be perturbed in limited ways. We overcome this problem by allowing for reward learning over repeated interactions with the agent.

Monderer and Tennenholtz [14] studied the problem of k-implementation, in which an interested party can assign positive monetary rewards to influence the actions of agents in games. The work assumes an interested party who can assign unlimited rewards to states and uses commitment to implement desirable outcomes with only minimal assumptions about agent rationality. The work provides a thorough analysis of the cost of implementing desired outcomes, but does not deal with learning agents' preferences.

The basic idea of inferring preferences by observing behavior is not new [6, 15, 19], but the idea of *revealed preference* is typically passive, and applied without modifying the environment to make further inferences. A dynamic approach is taken in the economics literature on Bayesian learning and experimentation, where one gathers additional preference information through experimentation and performs Bayesian updating to refine estimates and separate out confounding beliefs [3, 12]. The learning problem in these papers is considerably simpler than that studied here.

1.2 Outline

We formalize the policy teaching problem in Section 2, and provide a linear programming formulation for the case of known agent rewards in Section 3. We turn to consider the case of unknown agent rewards in Section 4, where we first present a general active, indirect elicitation method and then provide tractable elicitation strategies with desirable convergence properties. In Section 5 we consider a specific policy teaching problem in an ad network setting, and present preliminary experimental results in a simulated ad network to demonstrate the effectiveness of our methods. We provide extensions to handle partial observations and partial goals in Section 6, and consider strategic agents in Section 7. All proofs are omitted in the interest of space and will be made available in a longer version of the paper.

2. POLICY TEACHING

The policy teaching problem [21] considers an agent performing a sequential decision task with respect to an infinite horizon MDP $M = \{S, A, R, P, \gamma\}$, where S is a finite set of states, A is a finite set of possible actions, $R : S \rightarrow \Re$ is the reward function, $P : S \times A \times S \rightarrow [0, 1]$ is the transition function, and $\gamma \in (0, 1)$ is the discount factor. Given M, the agent's decision problem is to choose actions for each state to maximize the expected sum of discounted rewards. Let π denote a stationary policy, such that $\pi(s)$ is the action the agent executes in state s. Given a policy π, the value function $V^\pi(s) = R(s) + \gamma \sum_{s' \in S} P(s, \pi(s), s')V^\pi(s')$ captures the expected sum of discounted rewards from state s. Similarly, the Q function captures the value of taking an action a and following the 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 [16], an optimal policy π^* maximizes the Q function in every state, such that $\pi^*(s) \in \arg\max_{a \in A} Q^{\pi^*}(s, a)$. We assume the agent can compute an optimal policy of its MDP, and that its inherent reward R is persistent.

[1] There is also a small literature on principal-agent problems with hidden actions in computer science. Chuang et al. [7] consider incentive problems in regards to agency in peer-to-peer systems. Babaioff et al. [2, 1] studied issues of "combinatorial agency" in which actions have combinatorial effects on the payoff of the interested party. Feldman et al. [8] earlier studied the issue of hidden actions in the context of multi-hop routing.

We consider an interested party who knows S, A, P, and γ, and aims to induce a pre-specified target policy π_T. The interested party can influence the agent's reward function by providing incentives $\Delta : S \rightarrow \Re$. We assume the agent is *myopically rational* and follows its optimal policy given Δ.[2] We assume that Δ affects the agent's reward function linearly, such that the agent plans with respect to $M' = \{S, A, R+\Delta, P, \gamma\}$ in the modified environment. We assume for now that the interested party observes the agent's policy.

We assume the interested party can only provide limited incentives. We define the notion of admissibility:

Definition 1. An incentive function $\Delta : S \rightarrow \Re$ is *admissible* with respect to a policy π_T if it satisfies the following linear constraints, denoted *admissible*(Δ):

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

The incentive value $V_\Delta^{\pi_T}(s)$ in Definition 1 captures the total sum of expected discounted incentives provided to an agent following policy π_T starting from state s. The limited spending constraint limits the total incentives provided to D_{max} when the agent performs π_T from the *start* state.[3] The no punishment condition ensures that only bounded, positive incentives are provided, which seems quite fitting in many of the web domains that motivate this work.[4]

3. THE KNOWN REWARDS CASE

When the interested party knows the agent's reward function, the policy teaching problem is to find minimal admissible incentives that induce the desired policy π_T. We first define the concept of inverse reinforcement learning (IRL) [15] to capture the space of rewards that are consistent with a particular policy:

Definition 2. Given a policy π and $M_{-R} = \{S, A, P, \gamma\}$, let $\{R : R \in IRL^\pi\}$ denote the set of reward functions for which π is optimal for the MDP $M = \{S, A, R, P, \gamma\}$. Furthermore, for $\epsilon > 0$, let $\{R : R \in IRL_\epsilon^\pi\}$ denote the set of rewards for which π is uniquely optimal for M by a slack of at least ϵ, such that $Q^\pi(s, \pi(s)) - Q^\pi(s, a) \geq \epsilon$ for all $s \in S$, $a \in A \backslash \pi(s)$.

The policy teaching problem then aims to find incentives leading to a reward function that is consistent with the desired policy:

Definition 3. Policy teaching with known rewards. Given an agent MDP $M = \{S, A, R, P, \gamma\}$, target policy π_T, incentive limits D_{max} and Δ_{max}, and $\epsilon > 0$, if there exists an admissible Δ such that $(R + \Delta) \in IRL_\epsilon^{\pi_T}$, find such a Δ to minimize $V_\Delta^{\pi_T}(start)$.

[2]We are also able to handle situations in which the agent plans with respect to an almost constant reward, or where the agent almost plans optimally. We postpone discussion to a longer version of the paper.

[3]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.

[4]Alternative definitions of admissibility are possible as well. Our methods are not specific to a particular admissibility definition, so we won't pursue the issue further in this paper.

The definition requires the provided incentives to strictly induce the desired policy. This avoids scenarios in which an agent is indifferent among multiple optimal policies and may choose one other than that desired by the interested party. To solve this problem, we need to (1) locate the space of reward functions under which π_T is uniquely optimal and (2) find an admissible incentive Δ that maps the agent's reward into this space. We apply a well-known result from inverse reinforcement learning, which shows that the space of rewards consistent with a particular (uniquely) optimal policy is given by a set of linear constraints:

THEOREM 1. (Ng and Russell [15]) *Given a policy π and $M_{-R} = \{S, A, P, \gamma\}$, $R \in IRL^\pi$ satisfies:*

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

Furthermore, for $\epsilon > 0$, $R \in IRL_\epsilon^\pi$ satisfies

$$(\mathbf{P}_\pi - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_\pi)^{-\mathbf{1}} \mathbf{R} \succeq \epsilon \quad \forall a \in A \tag{2}$$

where $\mathbf{P}_\pi$ and $\mathbf{R}$ are the transition function with respect to π and the reward function written in matrix form, and $\mathbf{I}$ is the identity matrix.

This theorem leads directly to our result:

THEOREM 2. *The following linear program solves the policy teaching problem with known rewards:*

$$\min_{\Delta, R_T} V_\Delta^{\pi_T}(start) \tag{3}$$

$$R_T(s) - \Delta(s) = R(s) \quad \forall s \tag{4}$$

$$((\mathbf{P}_{\pi_\mathbf{T}} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi_\mathbf{T}})^{-\mathbf{1}} \mathbf{R_T})[\mathbf{s}] \succeq \epsilon \quad \forall s, a \in A \backslash a_1 \tag{5}$$

$$admissible(\Delta) \tag{6}$$

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

Theorem 2 shows that policy teaching with a pre-specified target policy can be solved in polynomial time, unlike value-based policy teaching which is NP-hard and needs a mixed integer program formulation [21]. Another practical advantage of this formulation is that the interested party need only specify the desired policy and not its value function.

The main disadvantage is that the specified policy may not be inducible given the limits on incentives, whereas value-based policy teaching finds the best inducible policy given the limits on incentives and thus will always return the best available solution. In cases where the interested party can specify a set of desired policies, we can solve the LP for each desired policy until we find a feasible incentive provision to a particular desired policy. If the set of desirable policies is small, the LP formulation will still provide computational and representational benefits.

4. THE UNKNOWN REWARDS CASE

In most situations, the interested party will not know the reward function of the agent. This leads to a new policy teaching problem:

Definition 4. Policy teaching with unknown agent reward. Consider an agent following a policy π with respect to an MDP $M = \{S, A, R, P, \gamma\}$. An interested party observes the agent's policy, and knows $M_{-R} = \{S, A, P, \gamma\}$ but not R. Given target policy π_T, incentive limits D_{max} and Δ_{max}, and $\epsilon > 0$, if there exists an admissible Δ for which $(R + \Delta) \in IRL_\epsilon^{\pi_T}$, find an admissible Δ and observe agent policy π' such that $\pi' = \pi_T$.

Algorithm 1 Active indirect elicitation

Require: agent policy π, desired policy π_T, $\epsilon > 0$
Variables R, R_T, Δ; constraint set $K = \emptyset$
Add $R \in \text{IRL}^{\pi}$, $|R(s)| \leq R_{max}\ \forall s \in S$ to K
Add $R_T \in \text{IRL}_{\epsilon}^{\pi_T}$, $\Delta = R_T - R$ to K
Add $admissible(\Delta)$ to K
loop
    Find $\widehat{\Delta}$, $\widehat{R}$, $\widehat{R_T}$ satisfying all constraints in K
    if no such values exist **then**
        return FAILURE
    else
        Provide agent with incentive $\widehat{\Delta}$
        Observe π'
        if $\pi' = \pi_T$ **then**
            return $\widehat{\Delta}$
        else
            Add $(R + \widehat{\Delta}) \in \text{IRL}^{\pi'}$ to K

We assume that direct queries about the agent's preferences are unavailable, and that preference information must be inferred from observations of agent behavior. This is often true on the web; while firms such as Amazon and Facebook can observe user actions, it may be considered intrusive for them to ask their users for preference information, both because it disrupts from the user experience and because users may question their motives.

We adopt a simplified form of the active, indirect elicitation method from Zhang and Parkes [21], wherein the space of potential agent rewards is narrowed by drawing additional IRL constraints based on observations of agent behavior in response to provided incentives. We assume the agent's reward function is bounded in absolute value by R_{max} in every state, and maintain an 'IRL space' of reward functions that are consistent with observations *and* that have associated admissible incentive functions that can strictly induce the desired policy with some minimal slack $\epsilon > 0$. At every iteration, we make a guess $\widehat{R}$ at the agent's true reward by choosing a point in the IRL space. If our guess is correct, providing the associated incentives $\widehat{\Delta}$ will strictly induce π_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. Furthermore, we know that $R + \widehat{\Delta}$ induces π', which allows us to add the following IRL constraints to the IRL space:

$$(\mathbf{P}_{\pi'} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-1}(\mathbf{R} + \widehat{\mathbf{\Delta}}) \succeq \mathbf{0} \quad \forall a \in A \qquad (7)$$

IRL constraints contain $|S||A|$ constraints on R and restricts the space of possible rewards to the intersection of previous IRL space and the convex polytope implied by the added constraints. 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 we observe the desired policy or if the IRL space becomes empty.

We use the following notation for the algorithm. All constraints are added to a constraint set K, such that instantiations of variables must satisfy all constraints in K. An instantiation of a variable R is denoted as $\widehat{R}$. Algorithm 1 gives the elicitation method.

THEOREM 3. *Algorithm 1 terminates in a finite number of steps with a solution to the policy teaching problem with unknown rewards or returns FAILURE if no solution exists.*

Theorem 3 guarantees convergence regardless of the choice of $\widehat{R}$ and $\widehat{\Delta}$ from K in Algorithm 1. The minimal slack ϵ over the target policy ensures that all points within a closed hypercube of side length $\delta = \frac{\epsilon(1-\gamma)}{\gamma} - \kappa$ centered at $\widehat{R}$ are eliminated by IRL constraints whenever π_T is not observed, for some arbitrarily small $\kappa > 0$.[5] Since the true reward is consistent with IRL constraints, by a pigeonhole argument, only a finite number of such hypercubes of eliminated points can fit in the IRL space before elicitation converges.

In practice, the elicitation method is only useful if it can induce the desired policy after few agent interactions. We consider computationally tractable elicitation strategies that lead to few elicitation rounds. We present a centroid-based strategy that guarantees logarithmic convergence in Section 4.1, and a practical two-sided slack maximization heuristic in Section 4.2, adopted from [21] to this setting.

4.1 A centroid-based approach

Consider the IRL space at any round of the elicitation process. Since this set of reward functions is characterized by linear constraints, it is convex. We can apply the following result on cutting convex sets:

THEOREM 4. (Grünbaum [9]) *Any halfspace containing the centroid of a convex set in $\Re^n$ contains at least $\frac{1}{e}$ of its volume.*

By choosing the centroid of the IRL space of rewards for $\widehat{R}$, any added IRL constraint will cut off a constant fraction of the IRL space's volume:

LEMMA 1. *Let B_K^t denote the IRL space of reward functions implied by the constraints in K before the t-th iteration of Algorithm 1. Let c_t denote the centroid of B_K^t. Consider an elicitation strategy that picks $\widehat{R} = c_t$ and any corresponding admissible $\widehat{\Delta}$ for which $(\widehat{R} + \widehat{\Delta}) \in IRL_{\epsilon}^{\pi_T}$. Then providing $\widehat{\Delta}$ will either induce π_T or the added IRL constraints will eliminate at least $\frac{1}{e}$ of the volume of B_K^t, such that $vol(B_K^{t+1}) \leq (1 - \frac{1}{e})vol(B_K^t)$.*

Lemma 1 ensures that a constant fraction of the volume of the IRL space is cut off at each iteration. This implies that after a logarithmic number of iterations, the volume of the IRL space can be made arbitrarily small. If we can provide conditions under which the desired policy is elicited before the volume of the IRL space falls below some threshold, we can guarantee logarithmic convergence.

One condition that leads to logarithmic convergence is to ensure that all points within a small hypercube centered at the true reward are contained in the initial IRL space and never removed by added IRL constraints in cases where a solution exist. If points within this hypercube are chosen for $\widehat{R}$, the minimal slack over the target policy ensures that π_T is elicited. Under this condition, this suggests that we can stop the elicitation process after logarithmic rounds because we will either elicit the desired policy before the volume of the IRL space drops below the volume of the hypercube, or the true agent reward must not be contained in the initial IRL space and thus there are no possible solutions.[6] Unfortunately, this condition is not satisfied by Algorithm 1

[5]Throughout this paper, any mention of a hypercube refers to a closed, axis-aligned hypercube.

[6]Bertsimas and Vempala [4] used this general observation to formulate an algorithm for finding a point in a convex set specified by a separation oracle with logarithmic queries.

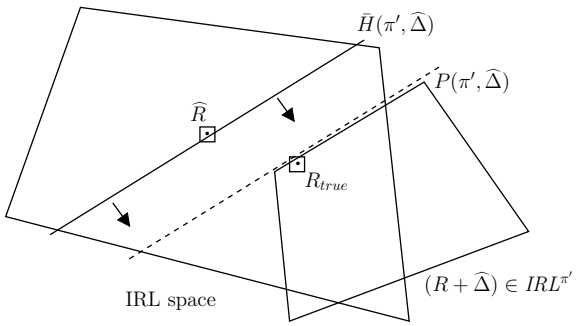

Figure 1: A condition that ensures logarithmic convergence requires a hypercube of points around the true reward R_{true} to be maintained throughout the elicitation process. The larger polyhedron in the figure represents the IRL space of rewards that have yet to be falsified. Given an observation π' based on incentives $\widehat{\Delta}$, the IRL constraints $(R + \widehat{\Delta}) \in IRL^{\pi'}$ represented by the smaller polyhedron may eliminate some points within the hypercube of points centered at R_{true}. To avoid this, we find a separating hyperplane $P(\pi', \widehat{\Delta})$ between the hypercube centered at $\widehat{R}$ and the IRL constraints, and shift $P(\pi', \widehat{\Delta})$ towards $\widehat{R}$ until it is arbitrarily close to $\widehat{R}$. The resulting hyperplane $\bar{P}(\pi', \widehat{\Delta})$ separates $\widehat{R}$ and the hypercube centered at R_{true}. Adding the corresponding halfspace $\bar{H}(\pi', \widehat{\Delta})$ instead of the IRL constraints ensures logarithmic convergence.

because IRL constraints may eliminate some points in the small hypercube centered at the true reward R_{true}. For a reward guess $\widehat{R}$ and associated incentive $\widehat{\Delta}$, the observed policy π' will be optimal for R_{true} but may not be optimal for all reward functions in the hypercube centered at R_{true}.

Nevertheless, we can modify our current algorithm to ensure that a hypercube of points centered at R_{true} are never eliminated. To do this, we adopt a technique introduced for the general environment design problem [20]. Since Theorem 3 ensures that all points within a closed hypercube of side length δ centered at $\widehat{R}$ are eliminated by added IRL constraints, by convexity there exists a separating hyperplane between this hypercube and the IRL constraints. Let $P(\pi', \widehat{\Delta})$ be such a separating hyperplane, and let $\bar{P}(\pi', \widehat{\Delta})$ denote a hyperplane that results from relaxing $P(\pi', \widehat{\Delta})$ in the direction perpendicular to itself until it is arbitrarily close to $\widehat{R}$. Let $\bar{H}(\pi', \widehat{\Delta})$ be the halfspace not containing $\widehat{R}$ that is defined by $\bar{P}(\pi', \widehat{\Delta})$. Since $P(\pi', \widehat{\Delta})$ separates R_{true} from a hypercube of side length δ centered at $\widehat{R}$, $\bar{P}(\pi', \widehat{\Delta})$ will separate $\widehat{R}$ from a hypercube of side length δ centered at R_{true}, as shown in Figure 1. This ensures that finding $\bar{H}(\pi', \widehat{\Delta})$ and adding it instead of IRL constraints will lead to logarithmic convergence.

In this setting, the hypercube of points centered at $\widehat{R}$ and the IRL constraints are both characterized by linear constraints. This allows us to find the separating hyperplane $P(\pi', \widehat{\Delta})$ by solving a simple linear program (e.g., see Theorem 10.4 of [18]). We can easily find $\bar{P}(\pi', \widehat{\Delta})$ by relaxing $P(\pi', \widehat{\Delta})$ until it almost passes through $\widehat{R}$ and define $\bar{H}(\pi', \widehat{\Delta})$ accordingly.

While this technique will lead to logarithmic convergence, it is a somewhat unsatisfying solution in that the found halfspace is a relaxation of the IRL constraints that may be much weaker than the full set of IRL constraints. One possible alternative is to relax each of the IRL constraints and use these relaxed constraints instead of $\bar{H}(\pi', \widehat{\Delta})$. Given observation π' based on $\widehat{\Delta}$, consider the following *relaxed IRL constraints* (rIRL):

$$(\mathbf{P}_{\pi'} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-1}[\mathbf{s}](\mathbf{R} + \widehat{\mathbf{\Delta}} + \mathbf{k_{sa}}) \geq 0 \quad \forall s, a \quad (8)$$

Here k_{sa} is a $|S|$-dimensional vector of parameter values whose i-th element is $\frac{\delta}{2}$ if $(\mathbf{P}_{\pi'} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi'})^{-1}[s][i] > 0$ and $-\frac{\delta}{2}$ otherwise. Given an observation π, we let $rIRL_\delta^\pi$ denote the relaxed IRL constraints.

This relaxation is sufficient to ensure that all points within a closed hypercube of side length δ centered at the true reward are not eliminated by relaxed IRL constraints. If any of the relaxed constraints eliminate $\widehat{R}$, then the intersection of all the relaxed constraints will also eliminate $\widehat{R}$ and we can thus add the entire set of rIRL constraints. However, it is possible to construct examples in which none of the relaxed constraints eliminate the reward guess $\widehat{R}$. This can only occur in situations where none of the individual IRL constraints eliminate the hypercube of points around $\widehat{R}$, which we expect to be fairly unlikely. Since it is easy to check whether $\widehat{R}$ is eliminated by plugging the point into the rIRL constraints, we propose an algorithm that adds rIRL constraints whenever possible and falls back to add $\bar{H}(\pi', \widehat{\Delta})$ instead if necessary.

In adapting the algorithm to achieve logarithmic convergence, we assume that the boundaries of the reward space are chosen such that the agent's true reward is bounded in absolute value by $R_{max} - \delta$ in every state. We define a modified version of Algorithm 1, denoted Algorithm 1*, where: (i) line 2 of Algorithm 1 adds $R \in rIRL_\delta^\pi$ instead of $R \in IRL^\pi$ to K, (ii) Algorithm 1 returns FAILURE if it has not returned after $1 + |S|\lceil \log_b \lceil \frac{R_{max}}{\delta} \rceil \rceil$ rounds, where $b = \frac{1}{1-k}$ for some k such that $0 < k < \frac{1}{e}$, and (iii) given observed policy π' based on $\widehat{\Delta}$, do not add $(R + \widehat{\Delta}) \in IRL^{\pi'}$ to K. Instead, check whether $(\widehat{R} + \widehat{\Delta}) \in rIRL_\delta^{\pi'}$. If not, add $(R + \widehat{\Delta}) \in rIRL_\delta^{\pi'}$ to K. Otherwise, find $\bar{H}(\pi', \widehat{\Delta})$ and add it to K.

THEOREM 5. *Assume the agent's true reward is bounded by $R_{max} - \delta$ in every state, where $\delta = \frac{\epsilon(1-\gamma)}{\gamma} - \kappa$ for some arbitrarily small $\kappa > 0$. For any elicitation strategy that picks the centroid of B_K^t for $\widehat{R}$, Algorithm 1* terminates with a solution to the policy teaching problem with unknown rewards or returns FAILURE if no solution exists after at most $1 + \lceil \log_b \lceil (\frac{R_{max}}{\delta})^{|S|} \rceil \rceil$ iterations.*

Since $\bar{H}(\pi', \widehat{\Delta})$ and added rIRL constraints eliminate the centroid of the IRL space while preserving a closed hypercube of points centered at the agent's true reward, the condition required for logarithmic convergence is satisfied and Theorem 5 follows. Here $(\frac{R_{max}}{\delta})^{|S|}$ is the number of non-overlapping hypercubes with side length δ that fit within the bounded space of rewards considered. This can be viewed as the size of the elicitation problem, and the bound given by Theorem 5 is logarithmic in this dimension. This logarithmic bound is still linear in the number of states though, because only one of the constraints added at each iteration is guaranteed to cut off a constant fraction of the volume.

Although computing the centroid exactly is #P-hard [17], polynomial time, randomized algorithms exist and extend

Grünbaum's result to the case of the approximate centroid. Bertsimas and Vempala [4] showed that any halfspace containing the average of $O(n)$ uniform samples from a convex set in $\Re^n$ will cut off a constant fraction of its volume with arbitrarily high probability. Using this result, we can construct an elicitation strategy that allows $\widehat{R}$ to be computed in polynomial time while guaranteeing logarithmic convergence with arbitrarily high probability:

THEOREM 6. *Assume the agent's true reward is bounded by $R_{max} - \delta$ in every state. For any elicitation strategy that picks the average of $O(|S|)$ points sampled uniformly from B_K^t for $\widehat{R}$, with arbitrarily high probability, Algorithm 1* terminates with a solution to the policy teaching problem with unknown rewards or returns FAILURE if no solution exists after at most $1 + \lceil \log_b \lceil (\frac{R_{max}}{\delta})^{|S|} \rceil \rceil$ iterations.*

Each iteration of Algorithm 1* with sampling is solvable in time polynomial in the number of states and actions. Since sampling $O(|S|)$ points uniformly takes $O(|S|^4)$ steps of a random walk that requires $O(|S|^2)$ operations per step, computing $\widehat{R}$ this way is $O(|S|^6)$ [4]. One can then find $\widehat{\Delta}$ satisfying $(\widehat{R} + \widehat{\Delta}) \in IRL_\epsilon^{\pi'}$ by solving a simple linear program.

4.2 Two-sided slack maximization heuristics

Despite the theoretical guarantees, a centroid-based approach may not scale well in practice for large state spaces. For this reason we also consider a heuristic approach to elicitation. We adopt a *two-sided slack maximization* heuristic defined for value-based policy teaching [21] to the present case of a fixed target policy, where we again obtain computational advantages over the value-based setting. From Theorem 3, we know that the minimal slack ϵ over the target policy provides a volume of points around an eliminated $\widehat{R}$ that are not the agent's true reward. If this volume of points is large and contained within the current IRL space, the added IRL constraints are guaranteed to make a large cut. The goal then is to perform a two-sided slack maximization to find a large volume of points around $\widehat{R}$ that are contained in the IRL space *and* that will be eliminated if the desired policy is not induced.

We consider the objective of choosing $\widehat{R}$ and $\widehat{\Delta}$ to maximize the minimal slack across all slack on the agent's initial policy π, induced policies π', and target policy π_T. We introduce a variable β to capture the minimal slack. We introduce variables α_s for all states s to capture the absolute value of the reward guess, and add a λ-weighted penalty term to the objective function. This allows us to express a preference for simpler rewards, and prevents the objective from choosing a large $\widehat{R}$ for the sake of increasing the slack. We have the following objective and constraints:

$$\max \beta - \lambda \sum_s \alpha_s \tag{9}$$

$$((\mathbf{P}_{\pi_\mathbf{T}} - \mathbf{P_a})(\mathbf{I} - \gamma \mathbf{P}_{\pi_\mathbf{T}})^{-\mathbf{1}} \mathbf{R_T})[s] \geq \beta \qquad \forall a, s$$

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

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

$$\alpha_s \geq R(s) \qquad \forall s$$

$$\alpha_s \geq -R(s) \qquad \forall s$$

We can use this heuristic to find $\widehat{R}$ and $\widehat{\Delta}$ in Algorithm 1 by solving a linear program containing the above equations and 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.

Many variants of the above heuristic are possible. For example, instead of using a shared slack β, one can introduce slack variables β_s for each state s, and maximize $\sum_s \beta_s$ in the objective. While the hypercube of points guaranteed to be eliminated is only provided through the smallest β_s, many points are likely to be eliminated when the slacks are large. To ensure a large volume, one can introduce a balancing variable $\mathcal{L}$ to capture the maximal absolute difference between any pair of β_s's, and add a θ-weighted term to the objective to penalize large differences. One obtains the following objective function and balancing constraint (other constraints remain as they were, with β_s replacing β):

$$\max \sum_s \beta_s - \lambda \sum_s \alpha_s - \theta \mathcal{L} \tag{10}$$

$$\beta_s - \beta_{s'} \leq \mathcal{L} \qquad \forall s, s' \in S, s \neq s'$$

We investigate these variations in the next section, where we consider a concrete example of a policy teaching problem in the ad network setting.

5. EXAMPLE: HYPERLINK DESIGN IN AN AD NETWORK

In an ad network, an ad provider serves advertising on a publisher's web site and the advertising revenue is shared between the two parties. An ad network provides a source of revenue for the publisher and ad provider, and matches web surfers to ads relevant to a web page's content. In 2008, Google generated $6.7 billion of revenue from its AdSense program, almost 80% of which were paid to publishers.[7]

Suppose that an ad provider wishes to affect the decisions of a publisher designing the hyperlink structure of a web site. The hyperlinks on web pages serve as paths for surfers to follow, and are designed to direct traffic as desired. From the publisher's perspective, a good hyperlink design directs surfers both to pages that generate high revenue, and to pages for which the publisher derives value from sharing the pages' contents. From the ad provider's perspective, a good hyperlink design directs surfers solely to pages that generate high ad revenue for the ad provider. By offering a larger share of the ad revenue on certain pages, the ad provider can try to induce the publisher to choose a more ad-effective hyperlink structure.

Immorlica et al. [10] provided a model of the publisher's hyperlink design problem.[8] For convenience, we adopt a simplified version of the publisher's decision model for the policy teaching problem. We define a set of states S where each state represents a page on the publisher's web site. Let $R(s)$ represent the publisher's utility for a surfer visiting page s. An action $a \in A$ selects a set of hyperlinks $H \subset S$ to place on a page. Choosing an action a in state s induces a probabilistic transition $P(s, a, s')$ that denotes the probability a surfer on page s clicks on a link to a page s', given

[7] Financial releases, http://investor.google.com

[8] The authors were concerned with characterizing the core of a cooperative game in which each web page on the publisher's web site is controlled by an autonomous agent. In their work there is no ad provider who is trying to influence the publisher's decisions, and no issues of unknown reward.

the hyperlinks on page s as selected by a. We introduce a dummy *start* state to indicate a surfer entering the web site, where $P(start, a, s)$ represents the probability of a user starting on a page s and is independent of a. We also introduce an absorbing terminal state s_E to indicate a surfer leaving the web site, where $P(s_E, a, s_E) = 1$ for all $a \in A$. We assume the probability that a surfer leaves the web site at any time step is constant, such that $P(s, a, s_E) = d$ for all $s \in S, a \in A$. The discount factor γ is set to $1 - d$ and represents the probability of a surfer staying on the web site. Without loss of generality we set $R(start) = R(s_E) = 0$. The publisher chooses an optimal hyperlink design π to maximize his expected sum of rewards from a surfer visiting the web site.

We consider an ad provider who observes the publisher's hyperlink design and knows $M_{-R} = \{S, A, P, \gamma\}$. These assumptions are reasonable because hyperlinks on web pages are public and traffic data is often provided to the publisher by the ad network. The ad provider does not know the publisher's reward function R, and cannot elicit this information directly. The ad provider can interact with the publisher multiple times, and aims to find an admissible incentive Δ to induce a desired hyperlink design π_T.

5.1 Simulated ad network

Assuming that a publisher is myopically rational and his reward function is persistent, we can apply the active indirect elicitation method to solve this policy teaching problem. As we are interested in seeing how our algorithm and elicitation heuristics may work in practice, we conduct preliminary experiments on a simulated ad network with the following parameters. We consider problem instances with 20 to 100 web pages and 5 actions. Each action a is generated to select from a specified set of 20 candidate pages, such that with probability $\frac{1}{2}$ a candidate page is one of the hyperlinks a places. We can view the candidate pages as the set of important pages on the web site, such that the publisher is deciding on which of these pages to include in the navigational links on each page. For each action a in state s, the transition function assigns positive probabilities to pages selected by a uniformly at random. These probabilities are scaled to sum to $\gamma = 1 - d = 0.7$, such that at any given time there is a 30% chance that a surfer leaves the web site.

The publisher's reward for each page is selected uniformly from [0,3]. Given the publisher's reward R, we find the interested party's target policy π_T by adding perturbations uniformly distributed from [0,1] to R and computing the optimal policy, discarding cases where the target policy matches the publisher's optimal policy. We set the minimal slack $\epsilon = 0.001$ and $\Delta_{max} = 10$, and set the incentive limit D_{max} by finding the minimal incentives necessary to strictly induce this policy by solving the linear program in Theorem 3 without the limited spending constraint.[9]

[9]With rewards defined over states, policies exist for which no reward functions can strictly induce these policies. To see this, consider an example of a 2 state MDP with actions stay and move. A policy which chooses to stay in both states implies a reward function with equal value for the two states. When reward functions are generalized to state-action pairs, every policy can be strictly supported by some reward function, e.g. by assigning positive rewards to pairs matching the agent's policy and 0 everywhere else. For the purposes of our experiment, we discard any generated instances for which no rewards strictly induce the target policy.

5.2 Experimental Setup

We perform simulations to evaluate the effectiveness of our elicitation method with various max-slack heuristics. We consider four different slack-based heuristics, corresponding to choosing $\widehat{R}$ and $\widehat{\Delta}$ to maximize the agent side slack, the target side slack, the two-sided slack, and the two-sided slack with a balancing term. We implement our methods in JAVA, using JOPT[10] as an interface to CPLEX version 10.1, which serves as our back-end LP solver. Experiments are conducted on a local machine with a Pentium IV 2.4Ghz processor and 2GB of RAM. We run 20 trials for each problem size and elicitation heuristic.

We need to set a few parameters for the algorithm and the elicitation heuristics. We set $R_{max} = 50$, which gives a large initial space of rewards that the algorithm will have to elicit over. For our max-slack heuristics, we wish to set the reward-penalizing coefficient small enough to avoid the objective function from choosing rewards with small slack and large enough to prevent choosing large rewards close to R_{max} for the sake of increasing the slack. To do this, we consider the max-slack objective function with only IRL constraints over the agent's initial policy, and perform a binary search to find the phase transition λ_0 where the solution reward is 0 everywhere for $\lambda > \lambda_0$ [15]. Since rewards just below λ_0 have zero reward in most states, they are likely to have small slack in many states. For our experiments then, we heuristically set $\lambda = \frac{\lambda_0}{2}$. Finally, for the variant of the max-slack heuristic with a balancing penalty term, we set $\theta = 0.75|S|$ to tradeoff between maximizing the minimal slack and the sum of slacks.

For all considered heuristics, each round of elicitation only requires solving a linear program, which only takes one to two seconds even for problems with 100 web pages. This presents a major improvement over value-based policy teaching [21], where computation time scaled exponentially in the number of states and problem instances with 20 states and 5 actions were already taking 4–7 minutes to solve.

5.3 Results

We plot the average number of elicitation rounds on different problem sizes for the four max-slack heuristics in Figure 2. In all cases considered, the elicitation process converged in fewer than 13 rounds on average. For most problem sizes, the two-sided slack with balancing objective averaged fewer rounds than the other heuristics, whereas the agent slack heuristic averaged more rounds than others.[11] In comparing the results to our theoretical bound, notice that the average number of elicitation rounds stays mostly constant as the number of states increases, whereas the logarithmic convergence bound is linear in the number of states. One possible explanation is that the IRL constraints are providing $|S||A|$ constraints per observation, as opposed to the one separating hyperplane used to establish the logarithmic convergence

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

[11]For each problem size, we conducted two-sample t-tests for the means for all pairs of heuristics. For most treatments, the difference in the average number of elicitation rounds is not statistically significant at the 5% level for a two-tailed test. There were a few treatments where the means between particular treatments were statistically significant, e.g., between the two-sided balancing and agent slack heuristics for 30, 50, and 80 states. However, more trials are required to obtain definitive results in this setting.

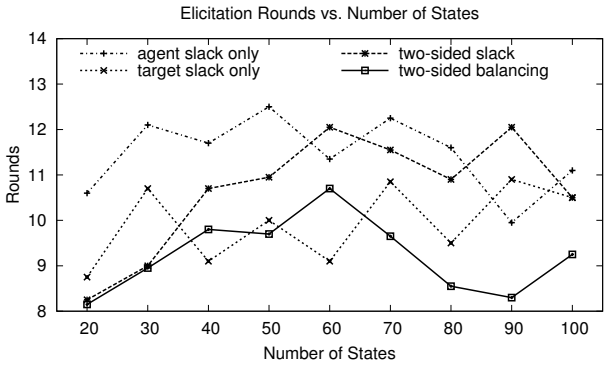

Figure 2: Convergence results for policy teaching on a simulated ad network. Results represent averages over 20 trials. The standard errors (not shown), averaged across all states, are 0.95, 0.70, 0.76, 0.68 for agent, target, two-sided, and two-sided balancing slack heuristics respectively.

bound. While our observation is only for this particular simulated setting, the results show some promise that the elicitation method and max-slack heuristics may be useful in practice.

6. HANDLING PARTIAL OBSERVATIONS AND PARTIAL TARGET POLICIES

We consider extensions of our model to handle situations where an interested party only observes the agent's actions in a subset of the states *and* may only wish to influence the agent's policy in a subset of the states. The results here also apply to situations of partial observations and full target policy, and vice versa. We envision a setting in which the interested party can only observe the agent's policy in a subset O of the state space S, such that the other states are private to the agent. Furthermore, instead of observing the agent's policy in all states $s \in O$, we assume the interested party only observes a trajectory of state-action pairs through states in $J \subset O$, where the set J depends both on the agent's policy and on the realization of the probabilistic transitions as the agent follows his policy.

In addition, we assume that the interested party's desired policy is only defined on a subset G of the state space S, such that the interested party specifies desired actions for states in G but is indifferent among agent actions in the other states. We denote a partial policy by π_T^G, such that the interested party aims to induce π_T^G even if some states in G are unobservable (not in O) or unobserved (not in J). We denote a partial observation by π^J, and assume that the interested party's goal is met as long as the actions in the observed states are as desired, such that $\pi^J(s) = \pi_T^G(s)$ for all $s \in J \cap G$.[12]

Partial observations and partial target policies present both computational and learning challenges. Computationally, we can no longer write down IRL constraints on observations and target policies because such constraints are defined with respect to a fully specified policy. For learning the agent's reward, our convergence results rely on eliminating a closed hypercube of points around a reward guess. It is not immediately clear whether partial observations based on a subset of the states is sufficient for the elicitation method to make progress. A similar problem exists for a partial target policy, where in providing incentives we aim to strictly induce the target policy in a subset of the states regardless of the agent's actions in other states.

To handle partial observations, we introduce binary variables X_{sa} over whether the agent took action a in state s. For any observation π^J and corresponding incentive $\widehat{\Delta}$, we can write down 'partial IRL' constraints that ensure the agent's Q values for observed decisions are higher than that of other actions, leaving unobserved decisions as variables:

$$Q(s,a) = R(s) + \widehat{\Delta}(s) + \gamma P_{s,a} V \qquad \forall s \in S, a \in A \quad (11)$$
$$V(s) \geq Q(s,a) \qquad \forall s \in S, a \in A \quad (12)$$
$$V(s) \leq M_v(1 - X_{sa}) + Q(s,a) \qquad \forall s \in S, a \in A \quad (13)$$
$$X_{sa} = o(s,a) \qquad \forall s \in J \quad (14)$$
$$\sum\nolimits_a X_{sa} = 1 \qquad \forall s \in S \quad (15)$$
$$X_{sa} \in \{0,1\} \qquad \forall s \in S, a \in A \quad (16)$$

Here Q, R, V, and X are variables. For all $s \in J$, $\underline{o(s,a)} = 1$ if $\pi^J(s) = a$ and $o(s,a) = 0$ otherwise; $M_v = \overline{M_v} - \underline{M_v}$, $\overline{M_v} = (\Delta_{max} + R_{max})/(1-\gamma)$, $\underline{M_v} = -R_{max}/(1-\gamma)$. Constraints 11–13 ensure that the agent's policy is optimal with respect to his true reward and the provided incentive $\widehat{\Delta}$. Constraint 14 assigns values to the binary variables for observed state-actions, and constraint 15 ensures that only one action is chosen per state. The Big-M constant M_v is set such that $V(s) = Q(s,a)$ for optimal actions and $V(s) \geq Q(s,a)$ otherwise. Note that the above constraints correspond to a particular observation, and we would introduce additional variables Q, V, and X for another observation, where the constraints from different observations are coupled only by the reward variable R.

At each round of elicitation, the interested party must choose $\widehat{R}$ and $\widehat{\Delta}$ such that $\widehat{R}$ is consistent with all past partial observations and $\widehat{R} + \widehat{\Delta}$ strictly induces the partial target policy. We can write 'partial IRL' constraints on the partial target policy similarly:

$$Q(s,a) = R(s) + \Delta(s) + \gamma P_{s,a} V \qquad \forall s \in S, a \in A \quad (17)$$
$$V(s) \geq Q(s,a) \qquad \forall s \in S, a \in A \quad (18)$$
$$V(s) \leq M_v(1 - X_{sa}) + Q(s,a) \qquad \forall s \in S, a \in A \quad (19)$$
$$\epsilon - \epsilon X_{sa} \leq V(s) - Q(s,a) \qquad \forall s \in G, a \in A \quad (20)$$
$$X_{sa} = t(s,a) \qquad \forall s \in G \quad (21)$$
$$\sum\nolimits_a X_{sa} = 1 \qquad \forall s \in S \quad (22)$$
$$X_{sa} \in \{0,1\} \qquad \forall s \in S, a \in A \quad (23)$$

Here Q, Δ, V, and X are variables specific to the partial target policy, and R is the reward variable shared with the partial IRL constraints from observed policies. For all $s \in G$, $t(s,a) = 1$ if $\pi_T^G(s) = a$ and $t(s,a) = 0$ otherwise. Constraint 20 ensures that the interested party strictly prefers desired actions for states in G.

[12] While some states are unobserved, the interested party may still be able to infer information about the agent's actions in these states based on knowledge about the agent's reward function. That said, we are only interested in eliciting desired actions in the observable states, whereas in principal-agent models the goal is often to elicit actions in hidden states.

The computational concerns are manifested in the use of binary variables and the need to move to a mixed-integer program from a linear program.[13] For the case of unknown rewards, we can still guarantee elicitation convergence:

THEOREM 7. *Consider the policy teaching problem with unknown rewards in a setting with partial observations and a partial target policy. Modify Algorithm 1 to add partial IRL constraints instead of IRL constraints, both for partial observations and the partial target policy. The elicitation process terminates after finite Δ adjustments with an agent performing the desired actions in the specified states that are observed, or returns FAILURE if no admissible Δ exists to strictly induce the partial target policy for some minimal slack $\epsilon > 0$.*

The key insight here is that when the reward guess and incentives are chosen to strictly induce a partial target policy, any partial observations to the contrary will falsify the reward guess and a volume of points around it. This is because points around the reward guess also strictly induce the partial target policy. These points may induce different actions in states not in G, but actions in those states are irrelevant. Given this, we can establish convergence as we had done for the standard model. Since partial observations and the partial target policy can be described solely with constraints, we can still choose any elicitation strategy for picking $\widehat{R}$ and $\widehat{\Delta}$.

The new learning challenge is manifested here in that we are no longer able to provide logarithmic convergence results. The problem is that the IRL space with the partial constraints is no longer guaranteed to be convex.

Remark. With partial observations it is possible for the agent to follow a trajectory consistent with the desired behavior but later follow another trajectory under the same incentives that reveals states in which the agent's policy is inconsistent with the desired policy. This poses no issues for the elicitation algorithm, as we can just provide the same incentives until we observe agent actions that differ from the partial target policy. In this sense the algorithm remains "open" while waiting for another Δ adjustment, because it may be that some state has not been traversed, and that this will occur with only very low probability. Once such states are traversed and an action is wrong, this presents new evidence and thus a new Δ is triggered.

7. HANDLING STRATEGIC AGENTS

Our discussion thus far relies on the assumption that an agent plans optimally with respect to the incentives currently provided and without consideration of the effect on future incentive provisions. While well motivated in many domains, we are interested also in handling *both* myopic and strategic agents in the same framework. A strategic agent may perform suboptimally in the current interaction in an attempt to distort the interested party's beliefs about his preferences. For example, a user who is willing to take a survey for a $5 gift certificate may nevertheless refuse to take the survey today if he believes that this refusal may lead to a better offer in the future.

[13]The technique of using binary variables to explicitly capture the agent's optimal policy is adopted from value-based policy teaching [21].

For the purpose of analysis, we consider an infinitely repeated game, in which each stage game describes the interested party's interaction with the agent. We assume that the interested party aims to induce the desired policy π_T and is indifferent with respect to the exact incentive provision as long as it is admissible. Furthermore, we assume that the agent knows the interested party's desired policy, and discounts values received in future interactions by a constant discount factor.

A standard approach to induce a strategic agent to choose a desired action is to use a *trigger strategy*, that provides maximal incentives as long as the agent performs the desired action but stops providing incentives as soon as any deviation is observed. This can be applied to the policy teaching problem; if an agent is sufficiently patient, he will perform π_T as long as doing so is better than his optimal policy without additional incentives. The interested party can immediately induce the desired policy, and without having to elicit underlying preferences.

However, this trigger-strategy approach does not satisfactorily solve the policy teaching problem. First, it is the threat to remove incentives in the future that induces the agent to follow π_T and the provided incentives need not make following π_T *myopically* optimal for the agent. This is outside the spirit of environment design, which aims to align the agent's optimal behavior in the *current environment* with that of the interested party and seeks to at least handle myopic planners. In extending to handle strategic agents, we do still want to also handle myopic agents. Second, the equilibrium based on the trigger strategy is not subgame perfect. Given the agent's policy, the interested party cannot commit to providing no future incentives because doing so implies that it will fail to implement the desired policy. Third, the trigger strategy does not allow any error from the agent's side and is thus quite a fragile solution.

Instead of pursuing a trigger strategy approach, we present an incentive scheme based on the active indirect elicitation method that overcomes these challenges and embraces both myopic and strategic agents. We consider a *sufficiently patient* agent, which means an agent that has a high discount factor for interactions with the interested party:

THEOREM 8. *Consider a strategic agent for whom there exists an admissible Δ such that $(R+\Delta) \in IRL_\epsilon^{\pi_T}$ for some $\epsilon > 0$. Consider an interested party who does not know R, follows Algorithm 1 when the IRL space is nonempty, and provides no incentives otherwise. There is a discount factor less than 1 with respect to interactions with the interested party for which the strategic agent will perform the desired policy before the IRL space becomes empty.*

This method guarantees that the interested party can induce the desired policy even if a strategic agent misrepresents his preferences. Compared with the trigger strategy approach, this approach works for both myopic and strategic agents. The key to the result is to recognize that it remains in the interest of a patient, but strategic agent to *eventually* receive incentives that induce the target policy when this is reachable through some admissible incentives. Thus, although the agent may misrepresent his rewards, he will not do so to the extent that this leads to a situation in which the target policy becomes unreachable, either because of inconsistencies due to an empty IRL space or because of inadmissibility due to the agent being too "greedy."

Although we do not argue that the strategy of the interested party forms an equilibrium with that of the agent, there is a sense in which the commitment issue is less severe than with the trigger approach. In particular, if the interested party adopts *myopic beliefs* – namely that the agent is acting in each round according to his underlying preferences – then the IRL space represents the space of agent rewards consistent with observations. Since the interested party selects incentives in each round that are optimal for some reward from this space, in this sense the interested party is always myopically best-responding to the agent.

Note that following the elicitation method does not guarantee that the interested party will achieve the target policy with a *minimal* incentive provision. Any incentives up to D_{max} may be provided, and this is true for the trigger strategy as well. In situations where the value of having the agent follow the desired policy greatly outweighs the specific costs of incentives, this is not a major issue. On the other hand, since Theorem 8 is based on the active indirect elicitation method, all our results on logarithmic convergence continue to hold and thus this method satisfies an interested party who is concerned about inducing the desired policy quickly.

8. CONCLUSIONS

The policy teaching problem is a specific example of the general problem of environment design, which is a powerful framework for action elicitation in environments with self-interested agents. We present a solution to the problem of finding limited incentives to induce a particular target policy, and provide tractable methods to elicit the desired policy after few interactions. Future work can look to extend this policy teaching framework by considering multiple agents, or learning (rather than planning) agents, adopting a Bayesian framework, or introducing alternative design levers beyond providing incentives. We would also like to further evaluate our algorithms in other settings to gain some insight into the elicitation process.

Acknowledgments

The authors gratefully acknowledge useful feedback from Jerry Green, Avi Pfeffer, Michael Mitzenmacher, and the anonymous reviewers on earlier versions of the paper. This work is supported in part by an NDSEG Graduate Fellowship to the first author.

9. REFERENCES

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

[2] M. Babaioff, M. Feldman, and N. Nisan. Mixed strategies in combinatorial agency. In *Proc. 2nd Int. Workshop on Internet and Network Economics (WINE'06)*, 2006.

[3] D. Bergemann and J. Valimaki. Learning and strategic pricing. *Econometrica*, 64(5):1125–49, September 1996.

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

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

[6] U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *Proc. 18th International Conf. on Machine Learning*, pages 35–42, 2001.

[7] J. Chuang, M. Feldman, and M. Babaioff. Incentives in peer-to-peer systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[8] M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-action in multi-hop routing. In *Proc. 6th ACM Conference on Electronic Commerce (EC'05)*, pages 117–126, 2005.

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

[10] N. Immorlica, K. Jain, and M. Mahdian. Game-theoretic aspects of designing hyperlink structures. In *Proc. 2nd Int. Workshop on Internet and Network Economics (WINE'06)*, pages 150–161, 2006.

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

[12] G. Keller and S. Rady. Optimal experimentation in a changing environment. *Review of Economic Studies*, 66(3):475–507, July 1999.

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

[14] D. Monderer and M. Tennenholtz. k-implementation. In *EC '03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 19–28, New York, NY, USA, 2003. ACM.

[15] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670, 2000.

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

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

[18] R. Vanderbei. *Linear programming : foundations and extensions*. Springer, 3rd edition, 2008.

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

[20] H. Zhang, Y. Chen, and D. Parkes. A general approach to environment design with one agent. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.

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

[22] H. Zhang and S. Zenios. A dynamic principal-agent model with hidden information: Sequential optimality through truthful state revelation. *Operations Research*, 56(3):681–696, 2008.