

Enabling Environment Design

via Active Indirect Elicitation

Haoqi Zhang and David Parkes

School of Engineering and Applied Sciences
Harvard University

7.13.08 / M-PREF

Agenda

- Why do people behave like they do?
- How can we get them to do something we (they) like?
- What are the relations among environment, behavior and preferences?

- Teachers want students to form effective study habits.
- Web 2.0 site wants a user to contribute content.
- Online retailer wants a customer to make purchases and write reviews.
- The government wants people to go green.

- Teachers want students to form effective study habits.
- Web 2.0 site wants a user to contribute content.
- Online retailer wants a customer to make purchases and write reviews.
- The government wants people to go green.

Often, the agent does not behave as desired by the interested party.

Why?

- Different preferences.
- Limited by personal and environmental constraints.
- Limited by allowable actions.
- Not limited enough by allowable actions.

- The environment affects the agent's behavior.
- If we change the environment we can indirectly affect the agent's behavior.

Environment Design

- An agent performs a sequence of observable actions in an environment.

Environment Design

- An agent performs a sequence of observable actions in an environment.
- An interested party modifies limited aspects of the environment.

Environment Design

- An agent performs a sequence of observable actions in an environment.
- An interested party modifies limited aspects of the environment.
- The agent may behave differently, but the interested party cannot otherwise impose actions.

Environment Design

- An agent performs a sequence of observable actions in an environment.
- An interested party modifies limited aspects of the environment.
- The agent may behave differently, but the interested party cannot otherwise impose actions.
- Goal: to induce desired behavior quickly and at a low cost.

Environment Design

- An agent performs a sequence of observable actions in an environment.
- An interested party modifies limited aspects of the environment.
- The agent may behave differently, but the interested party cannot otherwise impose actions.
- Goal: to induce desired behavior quickly and at a low cost.
- Focus on policy teaching (Z. & Parkes, 2008), where interested party can associate limited rewards with world states.

Understanding preferences

- Agent preferences often complex and unknown to interested party.

Understanding preferences

- Agent preferences often complex and unknown to interested party.
- Direct preferences elicitation is costly and intrusive.

Understanding preferences

- Agent preferences often complex and unknown to interested party.
- Direct preferences elicitation is costly and intrusive.
- Passive indirect elicitation (revealed preference) is insufficient.

Understanding preferences

- Agent preferences often complex and unknown to interested party.
- Direct preferences elicitation is costly and intrusive.
- Passive indirect elicitation (revealed preference) is insufficient.
- We introduce an active, indirect elicitation method.

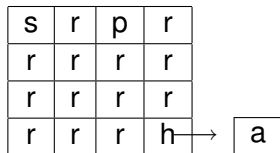
Markov Decision Process

Definition

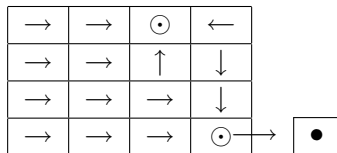
An **infinite horizon MDP** is a model $M = \{S, A, R, P, \gamma\}$:

- S is the set of states.
 - A is the set of possible actions.
 - $R : S \rightarrow \mathbb{R}$ is the reward function.
 - $P : S \times A \times S \rightarrow [0, 1]$ is the transition function.
 - γ is the discount factor from $(0, 1)$.
-
- We assume finite state and action spaces.
 - We also assume bounded rewards: $|R(s)| < R_{max}$ for all $s \in S$.

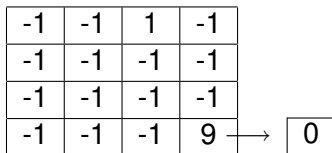
Child walking home from school



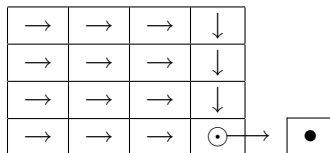
(a) state space



(b) child's policy



(c) child's reward



(d) parent's desired policy

Policy Teaching with known rewards

Policy Teaching with known rewards

- Agent performs optimal policy π that maximizes expected discounted sum of rewards.

Policy Teaching with known rewards

- Agent performs optimal policy π that maximizes expected discounted sum of rewards.
- Interested party can provide **admissible** incentive $\Delta : S \rightarrow \mathbb{R}$ (e.g., limited expected spending and no punishment).

Policy Teaching with known rewards

- Agent performs optimal policy π that maximizes expected discounted sum of rewards.
- Interested party can provide **admissible** incentive $\Delta : S \rightarrow \mathbb{R}$ (e.g., limited expected spending and no punishment).
- Agent performs π' w.r.t. $R + \Delta$.

Policy Teaching with known rewards

- Agent performs optimal policy π that maximizes expected discounted sum of rewards.
- Interested party can provide **admissible** incentive $\Delta : S \rightarrow \mathbb{R}$ (e.g., limited expected spending and no punishment).
- Agent performs π' w.r.t. $R + \Delta$.
- **Goal**: provide (minimal) admissible Δ to induce π_T .

Policy Teaching with known rewards

- Agent performs optimal policy π that maximizes expected discounted sum of rewards.
- Interested party can provide **admissible** incentive $\Delta : S \rightarrow \mathbb{R}$ (e.g., limited expected spending and no punishment).
- Agent performs π' w.r.t. $R + \Delta$.
- **Goal**: provide (minimal) admissible Δ to induce π_T .

Question: what reward functions induce π_T ?

Inverse Reinforcement Learning (IRL)

- Revealed preference for MDP
- Space of rewards consistent with observed policy is characterized by linear constraints (Ng and Russell, 2000)

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

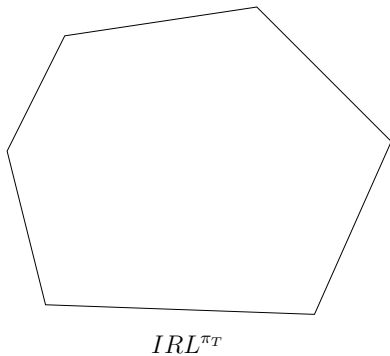
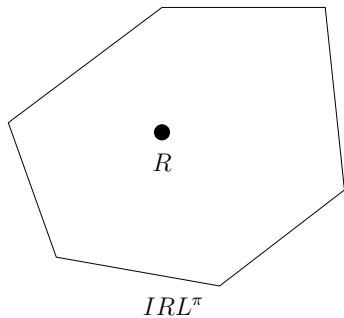
Policy teaching with known rewards

Solution via linear programming.

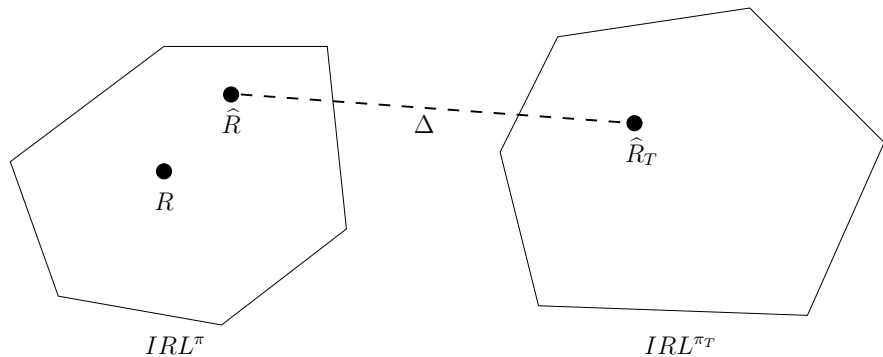
$$\begin{aligned} \min_{R_T, \Delta} V_{\Delta}^{\pi_T}(\text{start}) \\ R(s) + \Delta(s) = R_T(s) \quad \forall s \in S \\ (\mathbf{P}_{\pi_T} - \mathbf{P}_a)(\mathbf{I} - \gamma \mathbf{P}_{\pi_T})^{-1} \mathbf{R}_T \succeq \epsilon \quad \forall a \in A \setminus a_1 \\ \text{admissible}(\Delta) \end{aligned}$$

- Typically, the interested party won't know the agent's reward.
- Idea: provide incentives, observe behavior, and repeat.

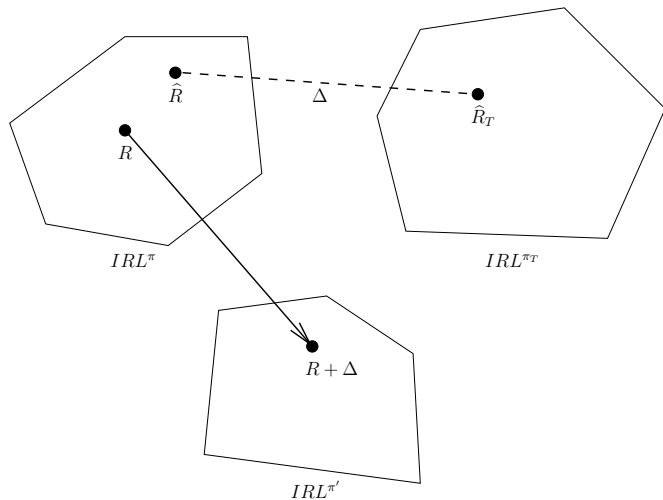
An indirect approach



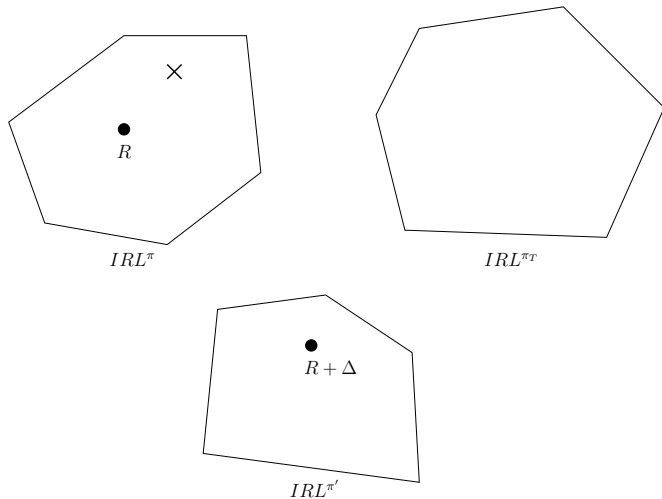
An indirect approach



An indirect approach



An indirect approach



Elicitation Algorithm

- Given policies π, π_T ; variables R, R_T, Δ ; constraint set $K = \emptyset$.
- Add $R \in IRL^\pi, R_T \in IRL_{\text{strict}}^{\pi_T}, \Delta = R_T - R, \text{admissible}(\Delta)$ to K .
- Repeat:
 - ▶ Find $\hat{\Delta}, \hat{R}, \hat{R}_T$ that satisfy K .
 - ▶ If no such values exist, return FAILURE.
 - ▶ Otherwise:
 - ★ Provide agent with incentive $\hat{\Delta}$.
 - ★ Observe agent policy π' w.r.t. $R' = R^{\text{true}} + \Delta$.
 - ★ If $\pi' = \pi_T$, return $\hat{\Delta}$.
 - ★ Otherwise add $(R + \hat{\Delta}) \in IRL^{\pi'}$ to K .

Convergence result

Theorem

The elicitation algorithm terminates in a finite number of steps with an admissible mapping Δ , or returns FAILURE if no such mapping exists.

Intuition.

Pigeonhole argument on number of hypercubes that can fit in the polyhedron. □

Choosing an objective function

- Tractable
- Few elicitation rounds

Centroid-based approach

Theorem

(Grünbaum, 1960) Any halfspace containing the centroid of a convex set in \mathbb{R}^n contains $\frac{1}{e}$ of its volume.

Centroid-based approach

Theorem

(Grünbaum, 1960) Any halfspace containing the centroid of a convex set in \mathbb{R}^n contains $\frac{1}{e}$ of its volume.

- Pick the centroid of the IRL space for R (and any Δ) at every iteration

Centroid-based approach

Theorem

(Grünbaum, 1960) Any halfspace containing the centroid of a convex set in \mathbb{R}^n contains $\frac{1}{e}$ of its volume.

- Pick the centroid of the IRL space for R (and any Δ) at every iteration
- By Grünbaum (1960) and a separating hyperplane argument, added IRL constraints eliminate at least $\frac{1}{e}$ of its volume.

Centroid-based approach

Theorem

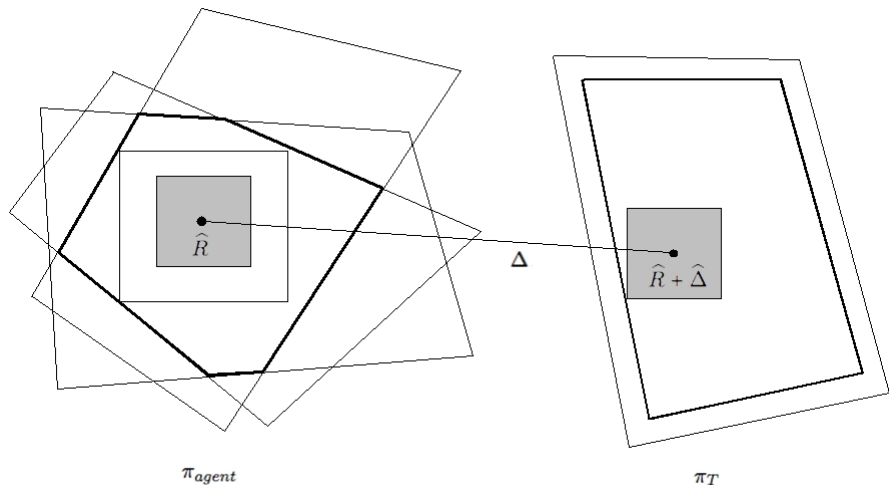
(Grünbaum, 1960) Any halfspace containing the centroid of a convex set in \mathbb{R}^n contains $\frac{1}{e}$ of its volume.

- Pick the centroid of the IRL space for R (and any Δ) at every iteration
- By Grünbaum (1960) and a separating hyperplane argument, added IRL constraints eliminate at least $\frac{1}{e}$ of its volume.
- Obtain **logarithmic** bound on the number of elicitation rounds.

- Centroid is #P-hard to compute (Rademacher, 2007).
- But can efficiently approximate centroid by sampling (Bertsimas and Vempala, 2004).
- Even better: any halfspace through the average of $O(|S|)$ uniformly at random samples cuts off a constant fraction of the volume of a convex set with arbitrarily high probability (Bertsimas and Vempala, 2004).

- Approximate centroid (in polynomial time) to obtain logarithmic bound with arbitrarily high probability.
- But approximation takes $O(|S|^4)$ steps of a random walk that takes $O(|S|^2)$ operations per step, so about $O(|S|^6)$.
- More tractable alternative?

Two-sided slack maximization heuristic



Maximize the minimal slack over *all* IRL constraints

(A linear program!)

Tidbit #1: Ad-network simulation

- Publisher designs link structure on website to maximize utility.
- MDP model (Immorlica et al., 2006)
- Policy teaching: an ad-network wishes to influence the publisher's link design to increase ad-revenue.
- Converges in few elicitation rounds:

Heuristic	Number of States				
	20	40	60	80	100
2-sided max slack	8.25	8.15	8.45	8.60	8.90
2-sided balancing	8.15	9.80	10.70	8.55	9.25
Target-slack only	8.75	9.10	9.10	9.50	10.50
Agent-slack only	10.60	10.90	10.85	11.25	11.50

Tidbit #2: Forward-looking agents

A strategic agent may misrepresent its preferences, but a simple teaching rule can nevertheless teach the desired policy when the agent is sufficiently patient and the behavior is attainable given the limited incentives.

Summary

- We discuss the problems of **Policy teaching** and **Environment Design**.
- We introduce a general active indirect **preference elicitation** method and provide desirable bounds on the number of elicitation rounds.
- We showed 2 tidbits.

Thank you

Please send comments and suggestions to hq@eecs.harvard.edu.