Repeated Inverse Reinforcement Learning

Kareem Amin*[1,2], Nan Jiang*[1], Satinder Singh[1] (*equal contribution)
[1]University of Michigan, [2]Google Research

Background

Big Question: how to *specify goals* (e.g., a reward function) for AI agents?

Challenges:

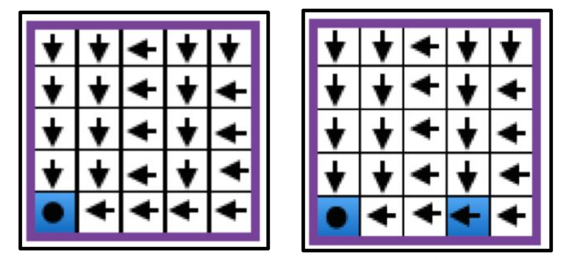
(1) **detailed** reward functions may be **difficult** to specify.

"[It] may be difficult to write down an explicit reward function specifying exactly how different desiderata should be traded off."

Pieter Abbeel & Andrew Ng [2]

(2) **simple** and abstract reward functions cause **value misalignment** (e.g., paperclip maximizer).

A Promising Approach: Inverse RL [1, 2]



Same policy explained by 2 reward functions

- Approach: infer the reward function from human *demonstration*.
- Success: can mimic a good policy in the environment (or *task*) of demonstration.
- Caveat: fundamentally **ill-posed**.
  There can be "**a large set** of reward functions for which the observed policy is optimal" (in a **single** task).
- Implication: no identification guarantee; **may not generalize** to new tasks.

Our Approach: consider **multiple** tasks (hence ***repeated*** IRL).

Problem Setup

Motivating Scenario: Value Alignment in AI Safety

- A task is specified as
  - A Markov environment $E = (S, A, P, \gamma, \mu)$. initial state distribution
  - A task-specific reward function R. (e.g., get to destination, make paperclips).
- Optimizing for R alone leads to unsafe AI. (e.g., ignore traffic lights, make gold paperclips).
- Assume θ_* captures safety concern / general preference that is **invariant** from task to task.
  (e.g., obey laws and social rules, be cost considerate)
  Human behavior π^* optimizes for $R + \theta_*$ in E.
- A sequence of tasks $\{(E_t, R_t)\}$ (share S, A, γ); agent receives demonstrations in **multiple** tasks.
- **Objective**: minimize the number of demonstrations.

Active Setting: Agent Chooses Tasks

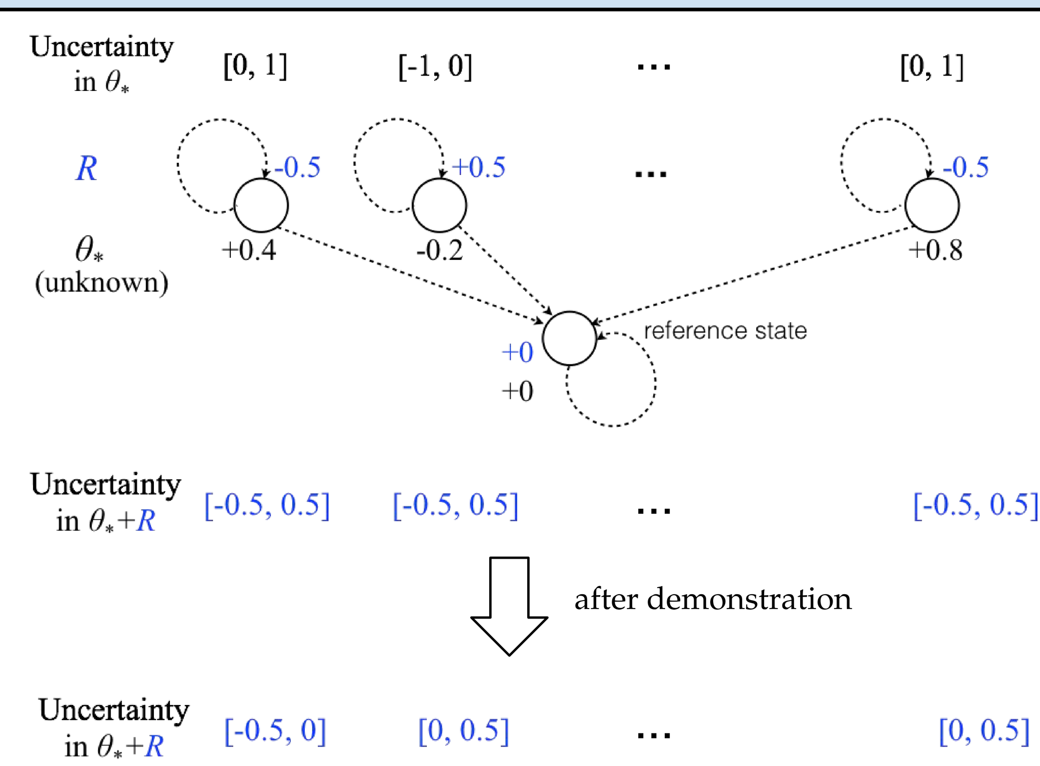
Protocol: for $t = 1, 2, \dots$

- Agent chooses (E_t, R_t).
- Human demonstrates π_t^* (optimal for $R_t + \theta_*$ in E_t).

Theorem: there exists an algorithm that outputs an θ s.t. $\|\theta - \theta_*\|_\infty \le \varepsilon$ after $O(\log(1/\varepsilon))$ tasks.

Note: identifying θ_* is literally impossible
- θ_* is *behaviorally equivalent* to $\theta_* + c\mathbf{1}$ (constant shift).
- To generalize: identifying the equivalence class is sufficient!
- Technically, we fix a reference state and assume θ_* to be 0 in that state.

Powerful identification but strong assumption



Passive Setting: Nature Chooses Tasks

Protocol: for $t = 1, 2, \dots$

- Nature chooses (E_t, R_t). Agent proposes π_t.
- If the loss of π_t is more than ε (i.e., a *mistake* is made), human demonstrates π_t^*.

$$\text{loss} = \mathbb{E}_{s\sim\mu}[V^{\pi^*}(s)] - \mathbb{E}_{s\sim\mu}[V^{\pi}(s)]$$

Issue: still ill-posed if nature never changes tasks.

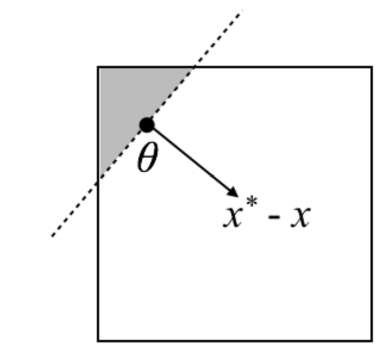
Resolution: address generalization directly -- prove upper bound on the number of mistakes.

Key idea: value is *linear* in rewards -- reduce to a linear bandit setting.

For MDPs, $d = |S|$ and each vector in X_t is the discounted occupancy of a policy.

Linear Bandit Protocol: for $t = 1, 2, \dots$

- Nature chooses (X_t, R_t), where $X_t \subset \mathbf{R}^d$, $R_t \in \mathbf{R}^d$. Agent proposes $x_t \in X_t$.
- If $\langle \theta_* + R_t, x_t \rangle < \langle \theta_* + R_t, x_t^* \rangle - \varepsilon$, human demonstrates x_t^*.

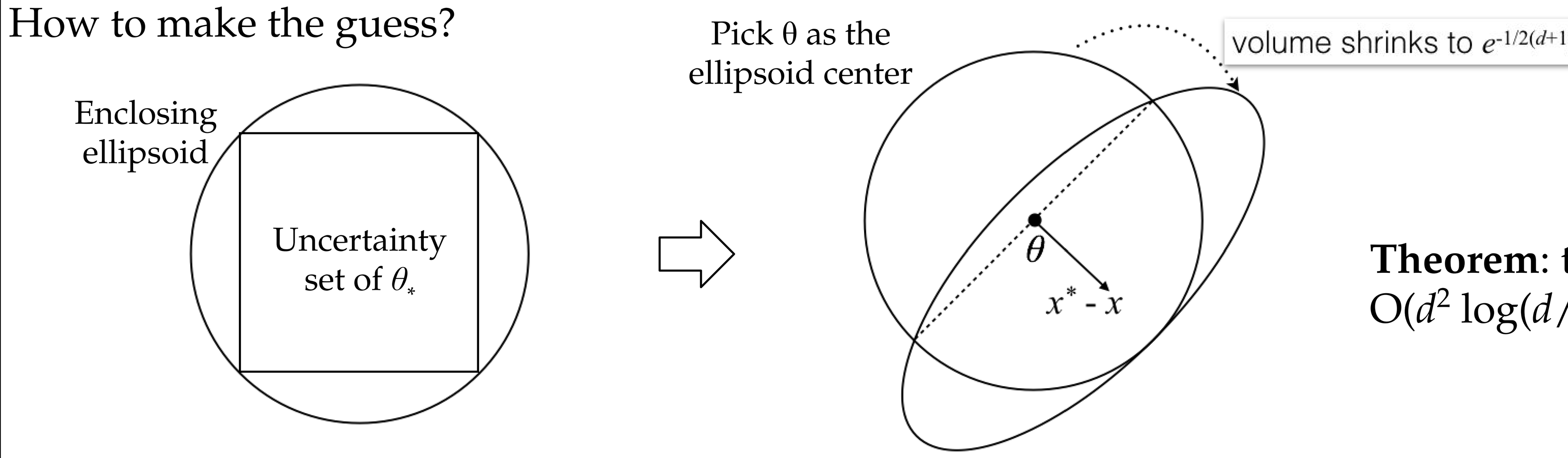


Algorithm & Analysis: pick x_t s.t. mistake leads to learning progress.

- Make a guess θ and behave greedily: $\langle \theta + R, x^* \rangle \le \langle \theta + R, x \rangle$
- When a mistake is made: $\langle \theta_* + R, x^* \rangle > \langle \theta_* + R, x \rangle$

$$\Rightarrow \langle \theta_* - \theta, x^* - x \rangle > 0$$

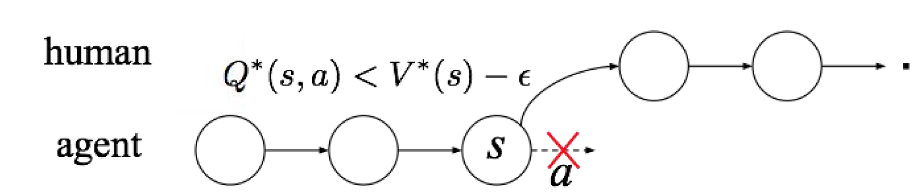
How to make the guess?



Theorem: the number of total mistakes is $O(d^2 \log(d/\varepsilon))$.

Trajectory-based protocol for the MDP setting

- Agent rolls out a trajectory.
- If a suboptimal action is chosen, human stops the agent.
- Human finishes the trajectory with an optimal policy.



Key idea: make an update in the ellipsoid algorithm after collecting a minibatch of mistakes.

Theorem: $\tilde{O}\left(\frac{d^2}{\epsilon^2}\log\left(\frac{d}{\epsilon\delta}\right)\right)$ total mistakes with probability at least $1 - \delta$.

More in the paper...

- **Lower bound (passive setting):** $\Omega(d \log(1/\varepsilon))$.
- **An intermediate setting:** agent chooses $\{R_t\}$ in a fixed environment E.
  - Identification guarantee depends on a diversity score of the environment.

References

[1] Andrew Ng and Stuart Russell. Algorithms for inverse reinforcement learning. ICML 2000.
[2] Pieter Abbeel and Andrew Ng. Apprenticeship learning via inverse reinforcement learning. ICML 2004.

https://arxiv.org/abs/1705.05427