Repeated Inverse Reinforcement Learning
Kareem Amin¹*², Nan Jiang¹*, Satinder Singh¹ (*equal contribution)
¹University of Michigan, ²Google 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.
(2) simple and abstract reward functions cause value misalignment (e.g., paperclip maximizer).

A Promising Approach: Inverse RL [1, 2]

- 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 $E = (S, A, P, \gamma, \mu)$.
- A task-specific reward function $R$.
- Optimizing for $R$ alone leads to unsafe AI.
- Assume $\theta^*$ captures safety concern / general preference that is invariant from task to task.
- Human behavior $\pi^*$ optimizes for $R + \theta$, in $E$.

- A sequence of tasks $(E_i, R_i)$ (share $S, A, \gamma$); agent receives demonstrations in multiple tasks.
- Objective: minimize the number of demonstrations.

Active Setting: Agent Chooses Tasks

Protocol: for $t = 1, 2, …$

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

Theorem: there exists an algorithm that outputs an $\theta$ s.t. $|\theta - \theta^*| \leq \epsilon$ after $O(d^4 \log(1/\epsilon))$ tasks.

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

Powerful identification but strong assumption

Passive Setting: Nature Chooses Tasks

Protocol: for $t = 1, 2, …$

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

Loss: 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.

Linear Bandit Protocol: for $t = 1, 2, …$

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

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

- Make a guess $\theta$ and behave greedily: $\langle \theta + R_t, x_t \rangle \leq \langle \theta + R_t, x^* \rangle$
- When a mistake is made:

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

How to make the guess?

Enclosing ellipsoid

Pick $\theta$ as the ellipsoid center

Theorem: the number of total mistakes is $O(d^4 \log (1/\epsilon))$.

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: $O(d^4 \log(1/\epsilon))$ total mistakes with probability at least $1 - \epsilon$.

More in the paper...

- Lower bound (passive setting): $\Omega(d \log(1/\epsilon))$

- Identification guarantee depends on a diversity score of the environment.

References


https://arxiv.org/abs/1705.05427