# Lecture 13: Inverse RL
# CS885 Reinforcement Learning

2022-10-31

Complementary readings:
Ziebart, B. D., Bagnell, J. A., & Dey, A. K. (2010). Modeling interaction via the principle of maximum causal entropy. In ICML.
Finn, C., Levine, S., & Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In ICML (pp. 49-58).

Pascal Poupart
David R. Cheriton School of Computer Science

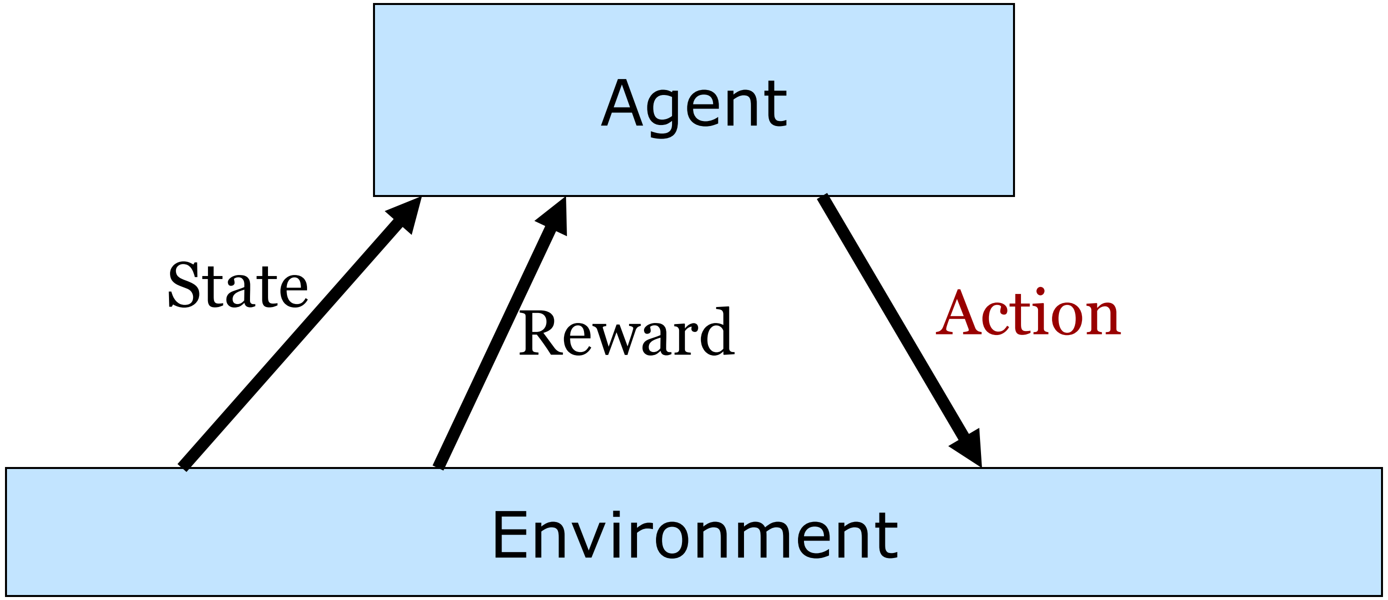
UNIVERSITY OF WATERLOO

# Outline

Inverse Reinforcement Learning (IRL)

- Feature expectation matching
- Maximum margin IRL
- Maximum entropy IRL
- Guided cost learning

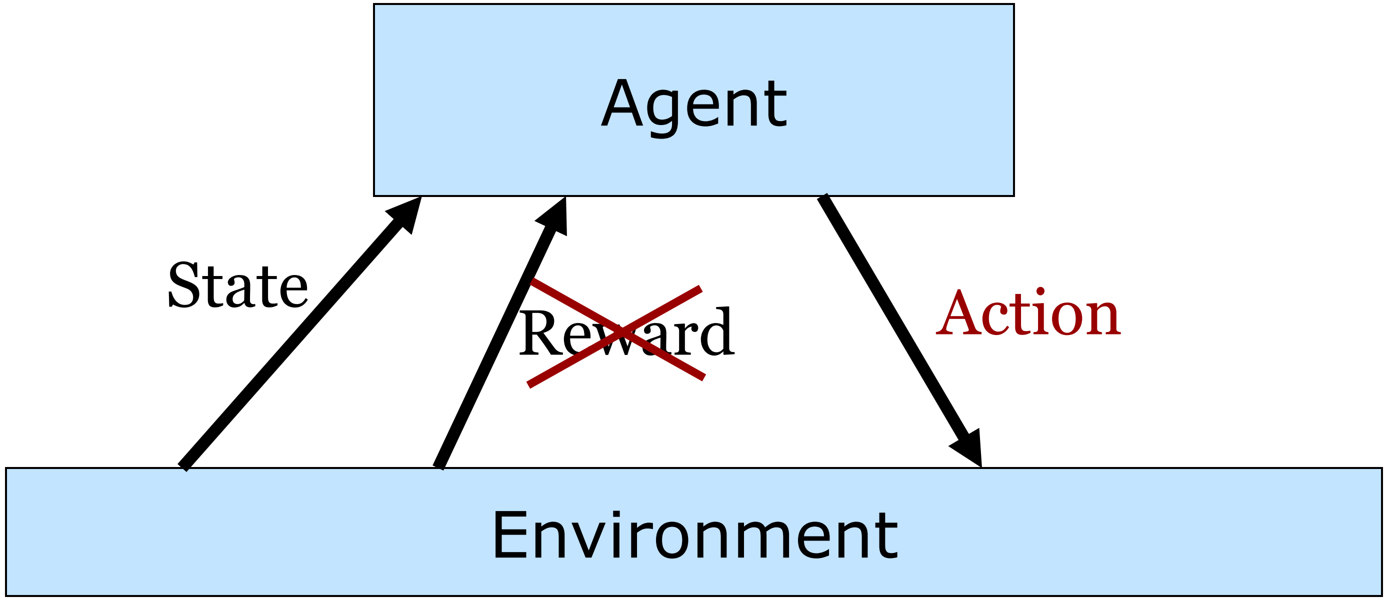
# Reinforcement Learning




**Goal:** Learn to choose actions that maximize rewards

$$s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_n, a_n, r_n \rightarrow \pi^*(a|s)$$

# Imitation Learning



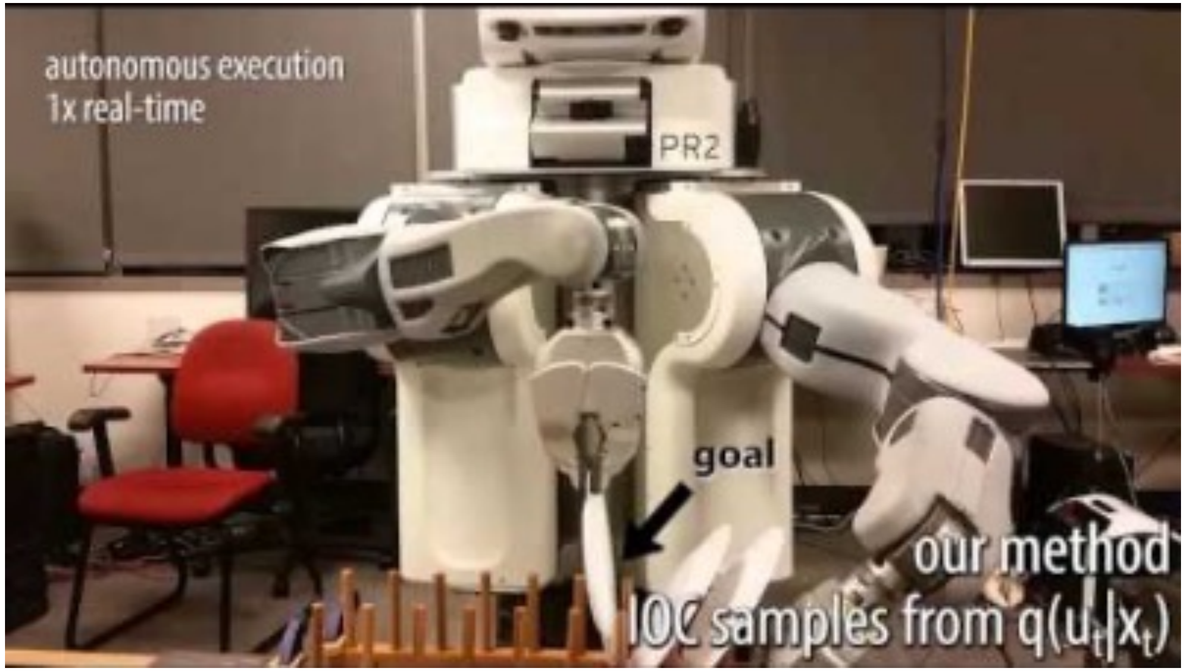

**Goal:** Learn to choose actions that imitate an expert policy

$$s_1, a_1^*, s_2, a_2^*, \ldots, s_n, a_n^* \rightarrow \pi^*(a|s)$$

# Problems with Imitation Learning

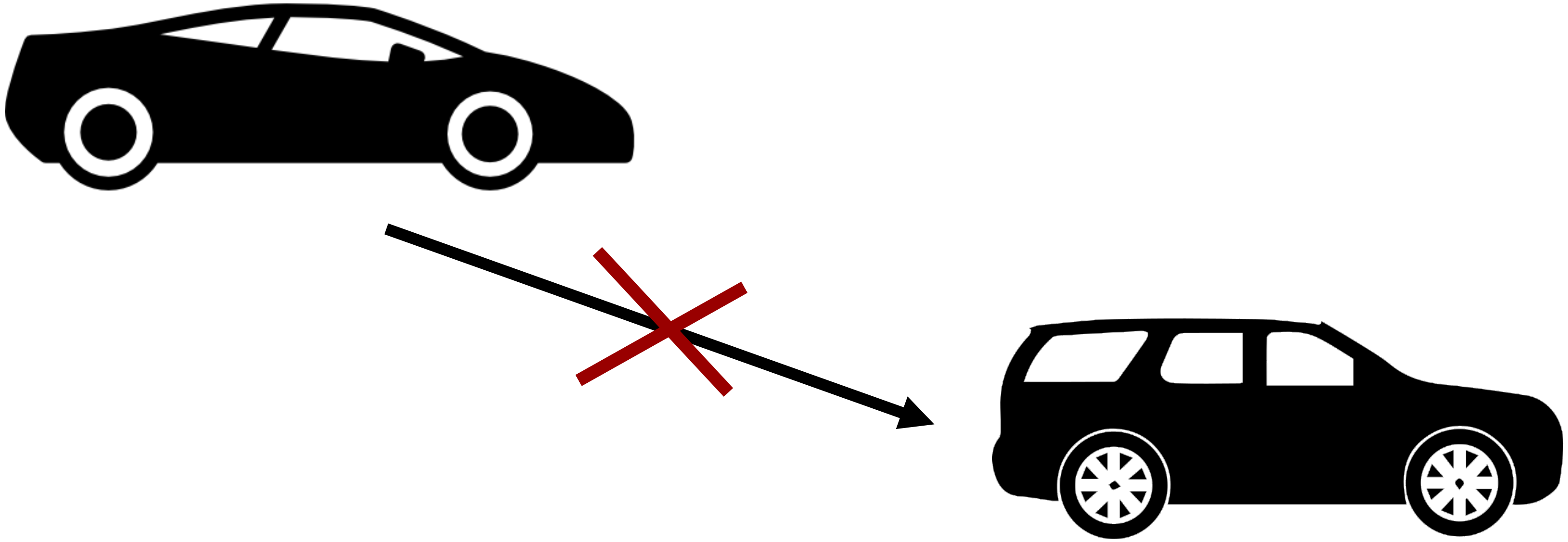
1. False assumption: state-action pairs are i.i.d. (independently and identically distributed

   - Non-smooth policy (effect on future states ignored)
   - Brittle policy (can't quantify how bad are suboptimal actions, errors may compound)



# Problems with Imitation Learning

2. Can't easily transfer what is learned to new domains



# Inverse RL

- Definition
  - States: $s \in S$
  - (Near) optimal actions: $a^* \in A$
  - Rewards: $r \in \mathbb{R}$
  - Transition model: $\Pr(s_t|s_{t-1}, a_{t-1})$
  - Reward model: $R(s, a) = E[r|s, a]$
  - Discount factor: $0 \leq \gamma \leq 1$
    - discounted: $\gamma < 1$ undiscounted: $\gamma = 1$
  - Horizon (i.e., # of time steps): $h$
    - Finite horizon: $h \in \mathbb{N}$ infinite horizon: $h = \infty$

unknown model

- Goal: find reward model $R(s, a) = E[r|s, a]$ such that

$$\pi^* = argmax_\pi \sum_{t=0}^{h} \gamma^t E_\pi[E[r_t|s_t, a_t]]$$

# General Approach

- Use IRL to learn reward function
- Then use reward function to learn policy
- Advantages:
  - No assumption that state-action pairs are i.i.d.
  - Transfer reward function to new environments

# Inverse RL Techniques

- Feature expectation matching
- Maximum margin IRL
- Maximum entropy IRL
- Guided cost learning
- Bayesian IRL
- Adversarial IRL

# Feature Expectation

- Assume that reward model $R(s,a)$ is a linear combination of some features $\phi_i(s,a)$:

$$R(s,a) = \sum_i w_i \phi_i(s,a) = \boldsymbol{w}^T \boldsymbol{\phi}(s,a)$$

- Value function:

$$\begin{aligned} V^\pi(s) &= \sum_t \gamma^t E_\pi[R(s_t,a_t)] \\ &= \sum_t \gamma^t E_\pi[\boldsymbol{w}^T \boldsymbol{\phi}(s_t,a_t)] \\ &= \boldsymbol{w}^T [\sum_t \gamma^t E_\pi[\boldsymbol{\phi}(s_t,a_t)]] \\ &= \boldsymbol{w}^T \overline{\boldsymbol{\phi}}^\pi \end{aligned}$$

# Feature Expectation Matching

- Idea: find weights $\boldsymbol{w}$ that define a reward model $R$ such that the optimal policy $\pi^{\boldsymbol{w}}$ (with respect to $R$ based on $\boldsymbol{w}$) matches expert feature expectation.
  - Let $\overline{\boldsymbol{\phi}^e}$ be the feature expectation of expert $e$
  - Let $\pi^{\boldsymbol{w}}$ be an optimal policy for $R = \boldsymbol{w}^T\overline{\boldsymbol{\phi}^e}$
  - Find $\boldsymbol{w}$ such that $\overline{\boldsymbol{\phi}^e} = \overline{\boldsymbol{\phi}^{\pi^{\boldsymbol{w}}}}$

- Problem: infinitely many $\boldsymbol{w}$ satisfy the feature expectation matching equality

# Maximum Margin IRL

Idea: find unique weights $\boldsymbol{w}$ that lead to the largest margin (value gap) possible between expert actions and non-expert actions.

- Let $\overline{\boldsymbol{\phi}^e}$ be the feature expectation of expert $e$
- Let $\pi^{\boldsymbol{w}}$ be an optimal policy for $R = \boldsymbol{w}^T\overline{\boldsymbol{\phi}^e}$
- Find $\boldsymbol{w}^* = \text{argmax}_{\boldsymbol{w}} \min_{\pi} \boldsymbol{w}^T(\overline{\boldsymbol{\phi}^e} - \overline{\boldsymbol{\phi}^\pi})$

# Maximum Margin IRL Pseudocode

Input: expert trajectories $\tau^e \sim \pi^{expert}$ where $\tau^e = (s_1, a_1, s_2, a_2, \dots)$
Estimate $\overline{\boldsymbol{\phi}^e}$ from $\tau^e$ and learn transition model $T$ from $\tau^e$
Initialize policy $\pi$ at random, sample $\tau \sim \pi$
Estimate $\overline{\boldsymbol{\phi}}$ from $\tau$ and initialize $\Phi = \{\overline{\boldsymbol{\phi}}\}$
Repeat

Compute weights that maximize margin

$$\boldsymbol{w}^* = argmax_{\{\boldsymbol{w}:\, ||\boldsymbol{w}||_2 = 1\}}\ margin\ \ s.t.\ \ margin \leq \boldsymbol{w}^T(\overline{\boldsymbol{\phi}_e} - \overline{\boldsymbol{\phi}}) \quad \forall \boldsymbol{\phi} \in \Phi$$

Compute optimal policy for $w^*$ :

$\pi^* = solveMDP(T, R, \gamma, h)$ where $R(s,a) = (\boldsymbol{w}^*)^T \overline{\boldsymbol{\phi}}$

Sample $\tau \sim \pi^*$, estimate $\overline{\boldsymbol{\phi}}$ from $\tau$ and update $\Phi \leftarrow \Phi \cup \{\boldsymbol{\phi}^*\}$

Until $margin \leq \epsilon$
Return $\boldsymbol{w}^*$ and $\pi^*$

# Issues with Maximum Margin IRL

- Maximizing the margin is arbitrary
- Problem: in some MDPs, the margin between the expert actions and some non-expert actions may be zero

- Idea: find the maximum entropy policy that matches the feature expectation of the expert
  - Benefit: naturally handles near but suboptimal actions in expert trajectories



UNIVERSITY OF WATERLOO

# Maximum Entropy IRL

Input: expert trajectories $\tau^e \sim \pi^e$ where $\tau^e = (s_1, a_1, s_2, a_2, \dots)$
Estimate $\overline{\boldsymbol{\phi}^e}$ from $\tau^e$ and learn transition model $T$ from $\tau^e$
Optimize weights:

$$\boldsymbol{w}^* = argmax_{\{\boldsymbol{w}: ||\boldsymbol{w}||_2 = 1\}} Entropy(\pi^*)$$

$$s.t. \quad \overline{\boldsymbol{\phi}^e} = \overline{\boldsymbol{\phi}^{\pi^*}}$$

$$\pi^* = solveSoftMDP(T, R, \gamma, h)$$

$$R(s,a) = \boldsymbol{w}^T \boldsymbol{\phi}(s,a)$$

Return $\boldsymbol{w}^*$ and $\pi^*$

# Limitations of Maximum Entropy IRL

- Applicability of vanilla Maximum Entropy IRL suffers from some limitations:
  - Linear reward model
  - Need to learn a transition model (model-based)
  - Need to solve MDP repeatedly

# Guided Cost Learning

- Extension of maximum entropy IRL to
  - Non-linear reward functions
  - Model-free techniques (no explicit transition model)
  - Iterative IRL (no repeated explicit MDP solving)

- See the following paper for details:
  - Finn, Levine, Abeel (2016) **Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization**, *ICML*.

# Demo (Guided Cost Learning)