Last Time:

Lecture 7

HRI, FALL '25

Andrea Bajcsy

- ☐ Boltzmann Rationality
- ☐ Intent Inference & Expression

This Time:

- ☐ Reward Learning
- ☐ Policy Learning

Reward learning vs. Policy learning?

It's often easier to <u>demonstrate</u> good behavior than to manually encode it into a (MDP) model.

<u>ex.</u> in autonomous driving it is hard to write down a reward function that models your personal driving "style."

Assume we are given some trajectory demonstrations:

$$\tau_D = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

Both cases of imitation learning

Inverse Reinforcement Learning (IRL)

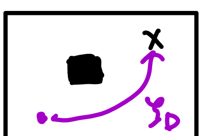
→ learn a reward function which, when optimized, yields a policy that does the task in the desired way

→ seeks to understand human demonstrator's objective, rather than just copying their actions

⊕ "interpretable" b/c you know "what matters" to the agent

⊕ can adapt/generalize to new environments (via RL)

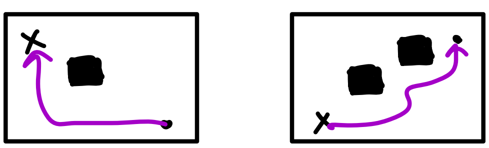
<u>ex.</u>



learn:

R = mind-the-■

Generalizes!



⊖ Typically needs xtra "RL steps"

Behavior Cloning (BC)

→ Directly learn a policy (π) that imitates the human demonstrator, mapping states → actions

→ Does <u>NOT</u> explicitly infer H demonst. underlying objective; just mimics a's

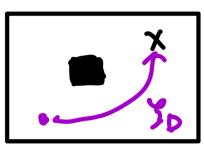
⊕ "simple" b/c you have the inputs (ex. states) and you have the "labels" (ex. actions); no need to infer reward.

⊕ computationally cheaper (no extra RL step)

⊖ compounding errors

⊖ struggles in unseen situations

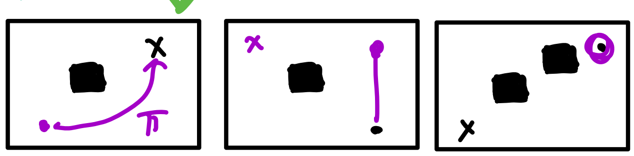
<u>ex.</u>



learn:

$\pi(a|s)$

Only imitates!



INVERSE REINFORCEMENT LEARNING – IRL (aka. Inverse Optimal Control)

Recall: "Forward" Reinforcement Learning

- Given: $s \in S$, $a \in A$, (sometimes) $P(s'|s,a)$, $r(s,a)$
- Goal: learn $\pi^*(a|s)$ s.t

$$\pi^* := \underset{\pi}{\operatorname{argmax}} \; \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t))\right]$$

over $P(s_{t+1}|s_t, \pi)$

INVERSE RL:

- Given: $s \in S$, $a \in A$, (sometimes) $P(s'|s,a)$ AND

$$\mathcal{D} := \{\xi_i\}_{i=1}^N \text{ demonstrations from } \pi^*$$

e.g. Humans!

- Goal: learn $r_\theta(s,a)$ ← (+ then use it to learn π^*)

reward params.

Q) Wait, but how do I search over all reward functions to find the "best" one given my dataset $\mathcal{D}$?

$\boxed{A1}$ Linear rewards are where a lot of foundations started b/c easier to optimize! Assume:

$$r_\theta(s) = \theta_1 \phi_1(s) + \theta_2 \phi_2(s) + \dots + \theta_d \phi_d(s)$$

$$= \theta^T \phi(s)$$

$\in \mathbb{R}$

feature function

$s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \in \mathbb{R}^n$

$\theta := \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \in \mathbb{R}^d$

feature vector: ex.

$\phi(s) := \begin{bmatrix} \text{distance to goal} \\ \text{distance to obstacle} \\ \vdots \end{bmatrix} \in \mathbb{R}^d$

MAX ENTROPY IRL [Ziebart et. al, AAAI 2008]

Relaxes the assumption that ξ_D is perfect, but it still allows the robot to learn H's reward.

Key idea is to treat demonstrations as observations drawn from some distribution that models the demonstrator as approximately optimal

SOUND FAMILIAR? :)

$$P(\xi \mid \theta) = \frac{e^{R_\theta(\xi)}}{\int e^{R_\theta(\bar{\xi})}\, d\bar{\xi}} \tag{1}$$

⇒ lets plug in our linear reward model to get:

$$P(\xi \mid \theta) = \frac{e^{\theta^T \phi(\xi)}}{\int e^{\theta^T \phi(\bar{\xi})}\, d\bar{\xi}} \quad (*)$$

Q How do we find θ (i.e. the reward parameter) given $\mathcal{D} = \{\xi_D\}$?

A1 If $\theta \in \{\theta^A, \theta^B, \theta^C, \ldots \theta^Z\}$ in (small) discrete, we could use Bayes' Rule to obtain $\hat{\theta}^* = \arg\max_\theta P(\theta \mid \mathcal{D})$ via lecture 5/6

A2 If $\theta \in \mathbb{R}^d$ is continuous, then its not tractable to have an exact posterior, then you compute instead the maximum likelihood estimate via gradient descent

$$\text{MLE:} \quad \hat{\theta} = \arg\max_\theta P(\xi_D \mid \theta)$$

$$\Leftrightarrow \quad = \arg\max_\theta \log \overset{(*)}{P(\xi_D \mid \theta)} \Rightarrow \max_\theta \log \frac{e^{\theta^T \phi(\xi_D)}}{\int e^{\theta^T \phi(\bar{\xi})}\, d\bar{\xi}}$$

$$= \max_{\theta} \boxed{\theta^T \phi(\xi_D) - \log\left[\int e^{\theta^T \phi(\xi)} d\xi\right]}$$

Know from opt. that we can find $\hat{\theta}$ via <u>gradient</u> <u>ascent</u>!

$$\nabla_\theta \boxed{} = (\text{algebra} + \text{calculus} \dots)$$

$$= \phi(\xi_D) - \mathbb{E}_{\xi \sim P(\xi|\theta)}\left[\phi(\xi)\right]$$

Max Ent IRL Gradient Ascent Rule:

$$\theta_{i+1} \leftarrow \theta_i + \alpha\Big(\underbrace{\phi(\xi_D)}_{\text{feature values from expert demos.}} - \underbrace{\mathbb{E}_{\xi \sim P(\xi|\theta_i)}\left[\phi(\xi)\right]}_{\text{expected feature values induced by reward param. guess } \theta_i}\Big)$$