Reinforcement Learning

Chapter 7: Applications and Advancements in DRL

Ali Bereyhi

ali.bereyhi@utoronto.ca

Department of Electrical and Computer Engineering

University of Toronto

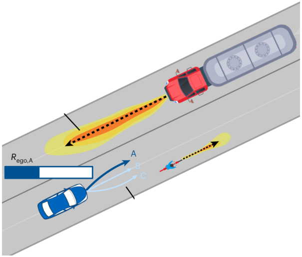
Fall 2025

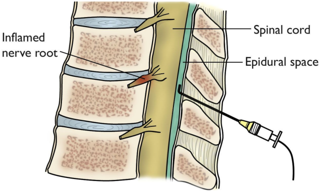
Challenge of *Sparse Rewarding*

In many RL problems, we deal with sparse rewards

- *In Frozen Lake game, we only get the reward at the end!*

Can this be tolerated in many practical problems?





Waiting for those sparse rewards could be very dangerous!

A Solution: *Reward Shaping*

If we could formulate what we want: maybe, we could shape the reward

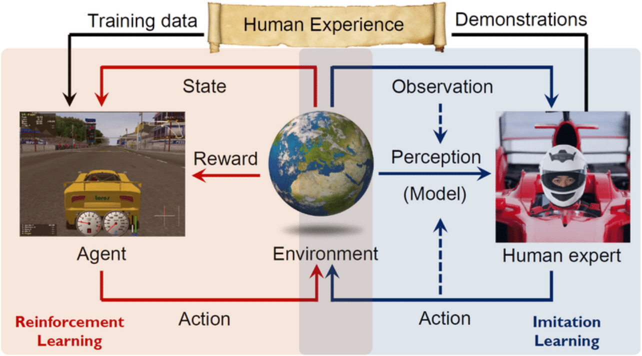
- *In Assignment 1, we did it for simple Frozen Lake game*
  ↳ *Just need to add some negative rewards in between*
- *By this approach, we make much denser reward*

+ *Can we always do it?*
– *Not really!*

In practice, we cannot necessarily formulate the types of rewards we want

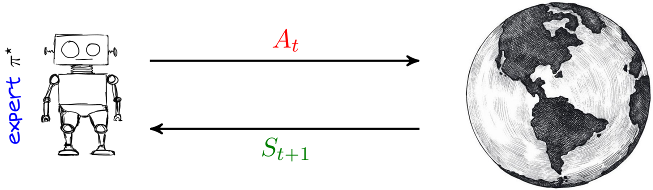
- *We often have access only to a human expert*
  ↳ *a professional driver or an expert surgeon*

Imitation Learning: *Learning by Expert*



In practice, we want to employ an expert to learn safely and efficiently

Behavior Cloning



We are looking for mimicking the expert

$$\min_{\boldsymbol{\theta}} \mathcal{L}\left(\pi_{\boldsymbol{\theta}}, \pi^\star\right)$$

and of course, we do it by sampling!

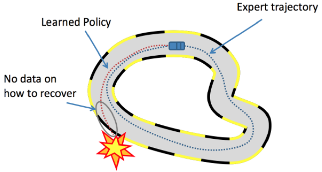
Behavior Cloning: *Not Always Efficient*

Algorithm 1: DAgger: Dataset Aggregation

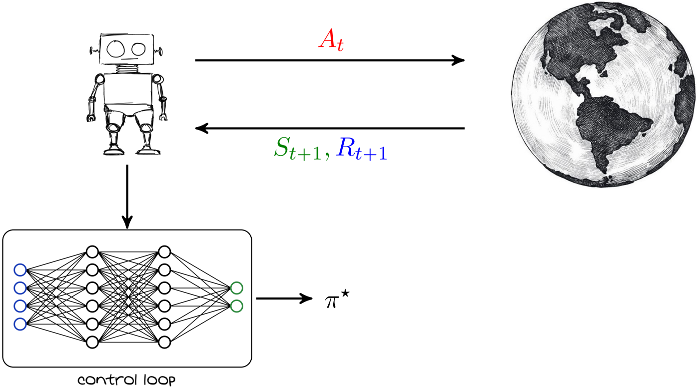
Data: π^*
Result: $\hat{\pi}^*$
$\mathcal{D} \leftarrow 0$
Initialize $\hat{\pi}$
for $i = 1$ **to** N **do**
  $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}$
  Rollout policy π_i to sample trajectory $\tau = \{x_0, x_1, \dots\}$
  Query expert to generate dataset $\mathcal{D}_i = \{(x_0, \pi^*(x_0)), (x_1, \pi^*(x_1)), \dots\}$
  Aggregate datsets, $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$
  Retrain policy $\hat{\pi}$ using aggregated dataset $\mathcal{D}$
return $\hat{\pi}$

DAgger: *famous efficient algorithm for behavior cloning*

But, we are always limited in collecting expert samples



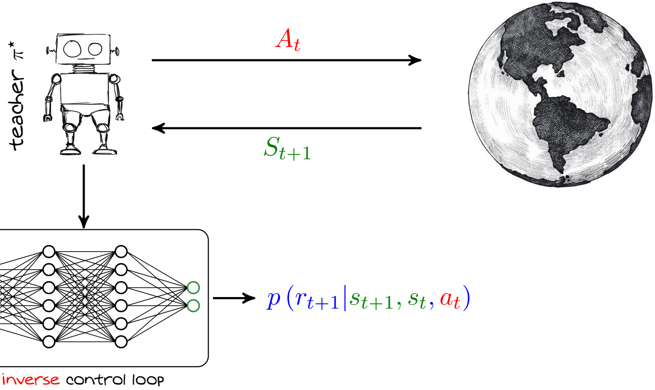
Inverse Reinforcement Learning



In classical RL setting

we collect rewards $\rightsquigarrow$ *we try to learn optimal policy*

Inverse Reinforcement Learning: *Learning Reward Function*



In inverse RL setting

a teacher plays $\rightsquigarrow$ we try to learn rewarding system

Inverse Reinforcement Learning: *Some Notes*

This problem can be cast using same formulation: we can write the optimal value function again

$$\begin{aligned} v_\star(s) &= \mathbb{E}_{\pi^\star}\left\{G_t | S_t = s\right\} \\ &= \mathbb{E}_{\pi^\star}\left\{\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \middle| S_t = s\right\} \end{aligned}$$

In RL, we know R_{t+i+1} and look for $\pi^\star$

- *We use function approximation, e.g., $\pi^\star(\cdot) \equiv f_{\mathbf{w}}(\cdot)$*

In inverse RL, we know $\pi^\star$ and look for R_{t+i+1}

- *We can again use function approximation, e.g., $p(r_{t+1}|\cdot) \equiv f_{\mathbf{w}}(\cdot)$*

Pieter Abbeel has a nice lecture on inverse RL: take a look at this link

Also, you may find this imitation learning lecture-note useful to learn a bit more!

Using Expert Demonstration

Not all experts can formulate their policy

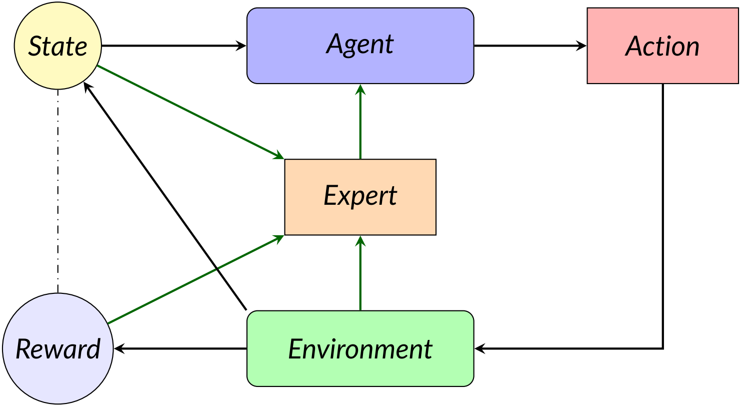
- *It might be not really easy to formulate it*
  - ↳ *A board-game player my act based on experience and intuition*
- *It might be much easier to demonstrate the policy*
  - ↳ *A professional driver could explain what they do in various conditions*

In this case learning the rewarding system is not really feasible

- *We don't exactly know optimal policy*
- *We can only describe it using expert demonstrations*

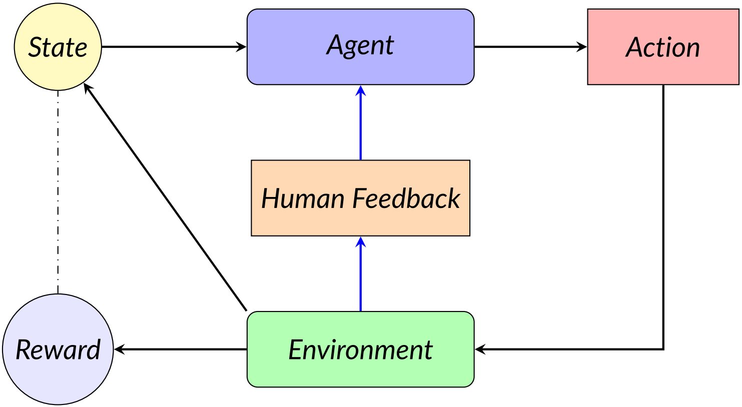
RL via *Expert Demonstration*

We could include expert demonstrations into our control loop



RLHF: *RL from Human Feedback*

This can be further extended to the use of human feedback: that human does not need to play optimal!



ChatGPT also extensively uses RLHF: see this lecture

Multi-Agent RL

Up to now, we considered other agents as a part of environment

- *But, they can have their own policies*
  - ↳ *They could collaborate to play jointly optimal*
  - ↳ *There might be adversary agents trying to impact negatively*

Take a look at this manuscript to learn about multi-agent RL

The End

Many thanks for attending the lectures . . .

- *Be confident and trust on what you have learned*
  - ↳ *You are now experts in Deep RL!*
- *Feel free to reach out*
  - ↳ *Would be more than happy to help!*
  - ↳ *I would appreciate your feedback on Quercus*
- *Always track the progress in the field*
  - ↳ *It's fast growing, so you should keep yourself posted*
  - ↳ *You are ready to follow any advanced topics in RL*

Next Summer, there will be a Generative AI course

↳ *We may use some Deep RL there as well* ☺