Advanced Topics in Reinforcement Learning

Lecture 21: Learning from Humans

Josiah Hanna
University of Wisconsin — Madison

Based on Slides from Stefano Albrecht's MARL lectures

Announcements

- Work on final projects.
- Read “Empirical Design in Reinforcement Learning”

Learning Outcomes

After today, you will be able to:

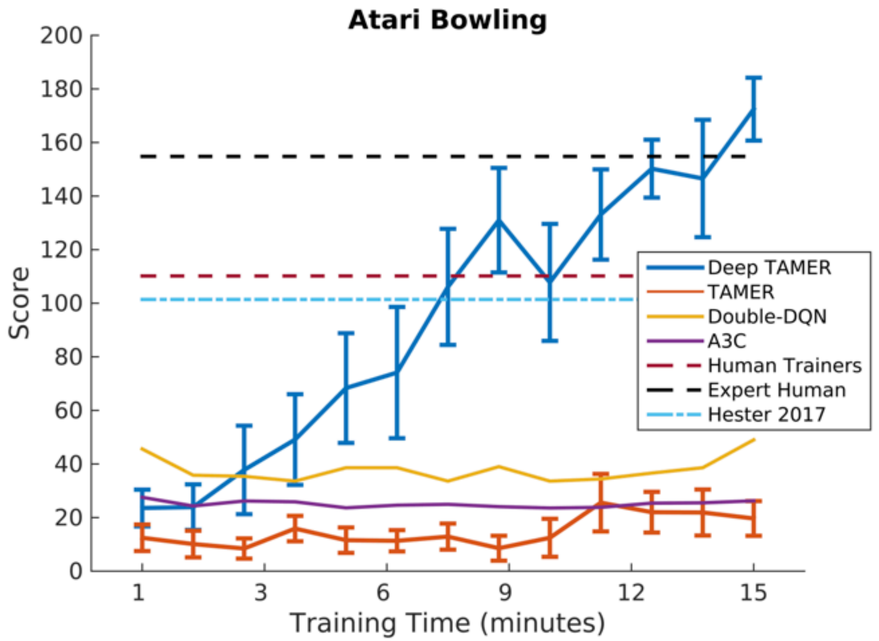
1. Compare and contrast imitation learning and inverse reinforcement learning.

2. Understand critical implementation details in the RLHF from preferences paper.

3. Formalize fine-tuning of LLMs as an RL problem.

Motivation

- RL can potentially yield decision-making that surpasses the ability of human decision-makers.
  - But, RL is slow and requires long training times even to match human ability on some tasks.
- Why not just learn from humans?



Warnell et al. (2018)

Imitation Learning

- Given data $\{(s_i, a_i)\}_{i=1}^{m}$, learn the policy that generated the data.
- $$\hat{\pi} = \arg\max_{\pi} \sum_{i=1}^{m} \log \pi(a_i \mid s_i)$$
- AKA behavior cloning.
- Strengths: learn from an expert, direct approach to distill expert knowledge.
- Weaknesses: requires an expert, basic approach requires action labels, lack understanding of *why* different actions were taken.

Deeper with Imitation Learning

- Supervised learning only considers 1-step errors.
- Leads to compounding error problem:
  - $J(\pi) \leq J(\pi^*) + \epsilon T^2$ (Ross and Bagnell 2010)

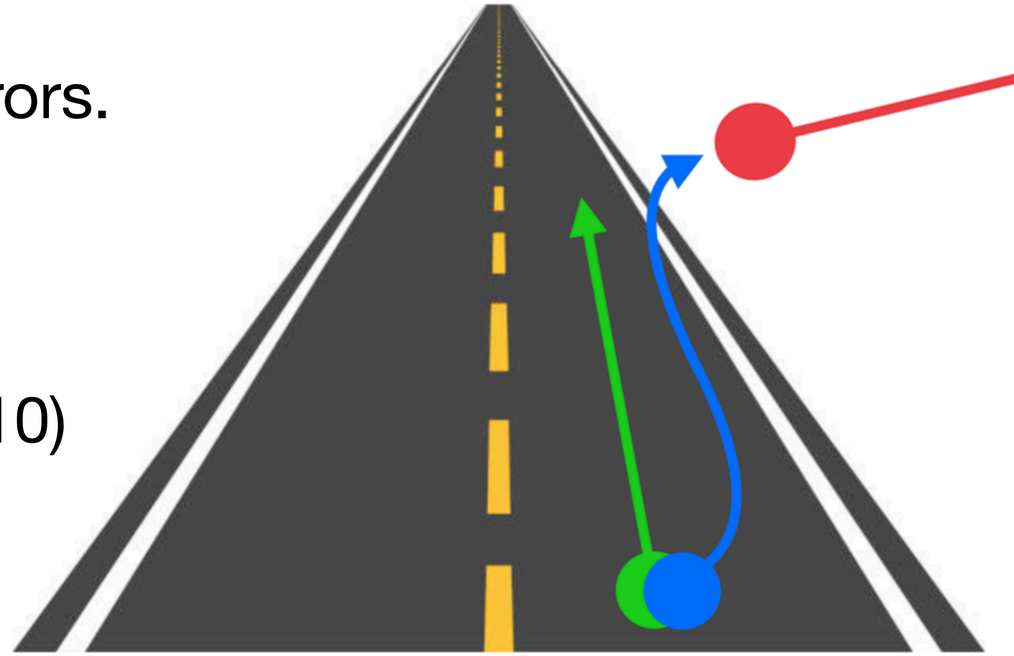


Image Credit: Scott Niekum

DAgger Algorithm

Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

Causal Confusion in Imitation learning

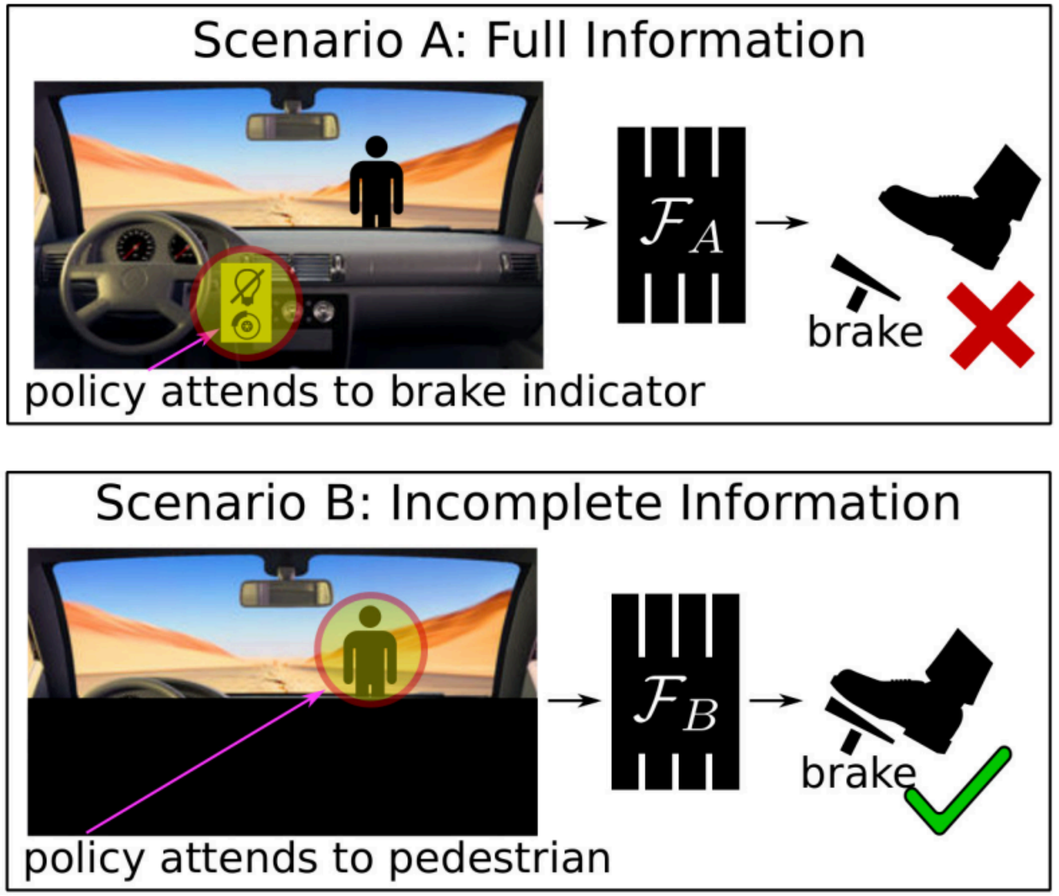


Image credit: Causal Confusion in Imitation Learning. De Haan et al. 2019.

Imitation Learning as Generative Modelling

- We can cast imitation learning as learning a generative model of expert behavior.
- Let d_π(s,a) be the joint probability on (s,a) and $d_*(s, a)$ be the expert's joint probability on (s,a).
- Select $\hat{\pi} = \arg\min_{\pi} D(d_* || d_\pi)$ where D is a (pseudo-) distance function on the state-action space (e.g., KL-divergence).
- Many techniques to solve this (GANs, diffusion models) but usually require some interaction with the environment during training.

GAIL (GANs for Imitation Learning)

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \,|\, s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \tag{18}$$

6: **end for**

Inverse Reinforcement Learning

- Given data $\{(s_i, a_i)\}_{i=1}^{m}$, learn a reward function, $\hat{r}$, such that the policy that generated the data is optimal w.r.t. $\hat{r}$.

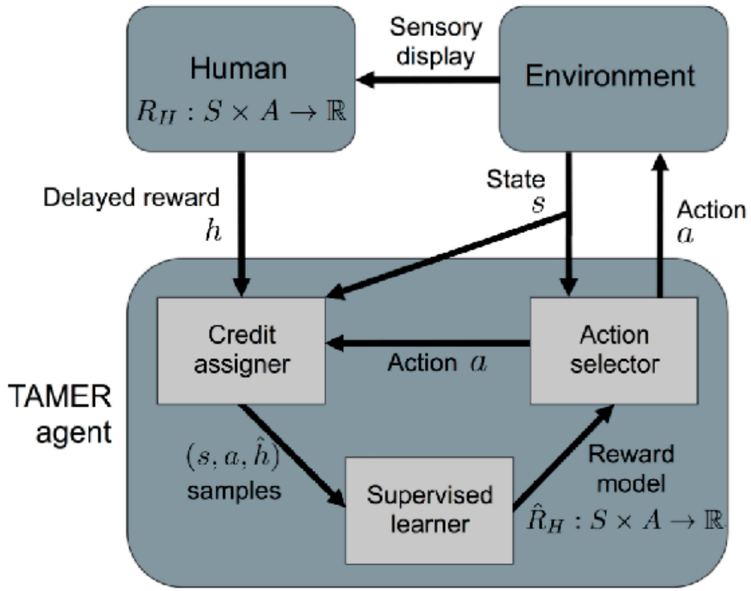
- Strengths: explains why actions were chosen, reward may be a more succint representation of a task.

- Weaknesses: ill-posed problem, assume expert is behavior optimally, indirect, traditionally has required solving an RL problem at each iteration.

RL from Human Feedback (RLHF)

- Perform RL as usual but also receive some type of feedback from a human observer. Convert this feedback to a reward signal for the agent.

- How to interpret scalar human feedback?
  - Good actions (reward)?
  - Good sequences of actions (return)?
  - *Comparatively* good sequences of action?
    - Regret



Knox (2012).

RLHF from Preferences

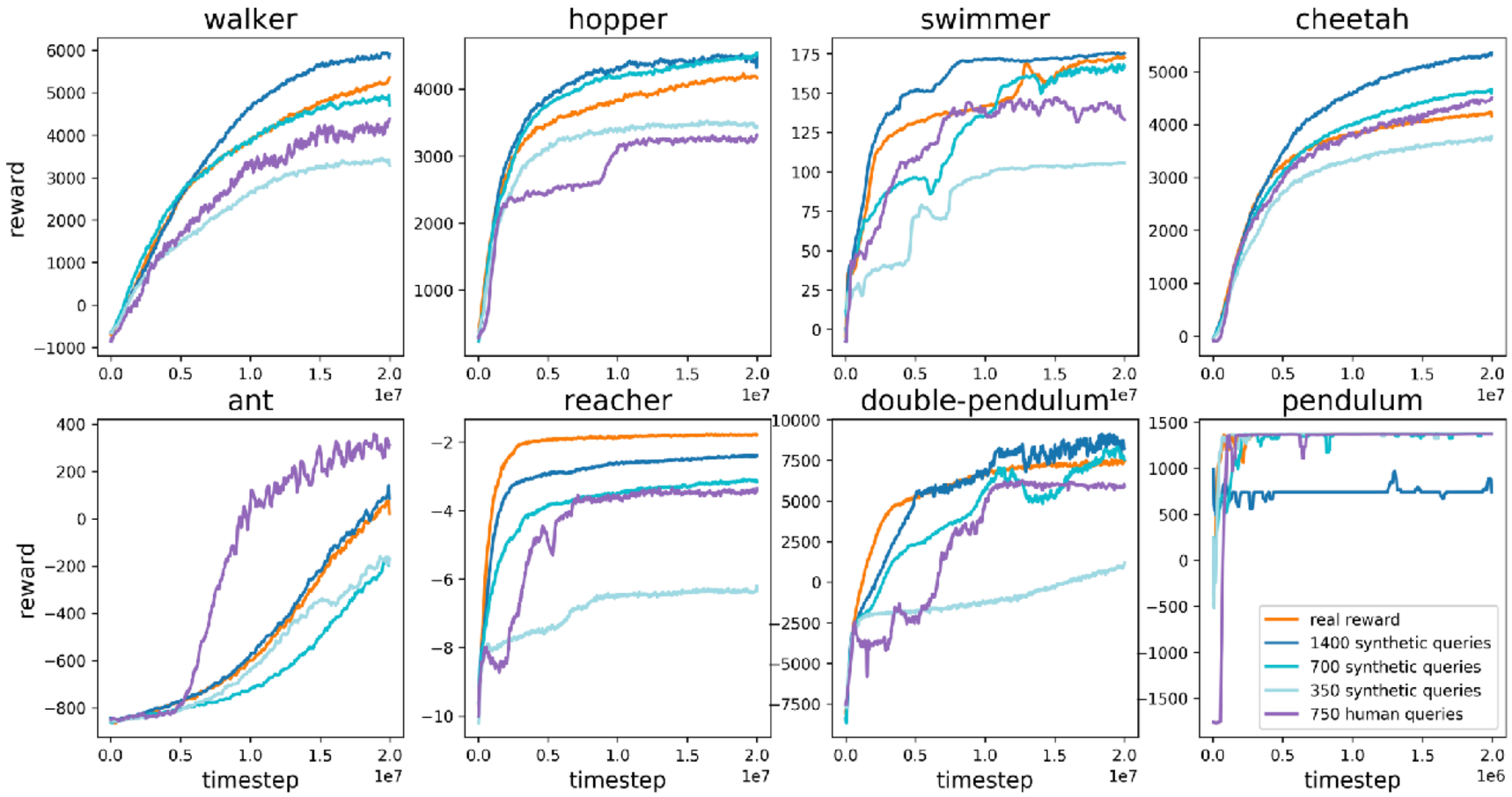
- Problem set-up: use human preferences over trajectory segments to infer underlying reward.
- Assume: $\Pr(\tau_1 > \tau_1) \propto \exp(\sum \hat{r}_\theta(s_t^1, a_t^1))$ where τ_1 and τ_2 are partial trajectories.
  - Bradley-Terry preference model (similar to Elo rating).
- Now we can fit $\hat{r}_\theta$ with maximum likelihood learning.

$$L(\theta) = \sum_{\tau_1, \tau_2} \mu(1) \log \Pr(\tau_1 > \tau_2) + \mu(2) \log \Pr(\tau_2 > \tau_1)$$

RLHF from Preferences

- Why infer a reward function in the first place?
- How to pair trajectories in the data?
- How often to query expert of rankings?
- Why normalize the reward?

RLHF from Preferences



Amr's Presentation

Training language models to follow instructions with human feedback

- Ouyang et al. (2022)
- Slides

RLHF for LLMs

- LLMs are autoregressive models of language or formally: $\pi : \mathscr{H} \rightarrow \mathscr{W}$.
- Can view as policies that map histories of text (i.e., context) to a probability distribution over next tokens to output.
- Often we discuss the probability of an entire response: $p_\theta(y \mid x)$ where x is a prompt and y is the full response that consists of a sequence of output tokens.

$$p_\theta(y \mid x) = \prod_{t=0}^{l} \pi_\theta(y_t \mid x, y_{0:t-1})$$

RLHF for LLMs

- Goal: given a reward function, $r(x, y)$, apply RL to maximize the probability of y in response to x.
  - Supporting goal: keep $p_\theta(y \mid x)$ close to the initial model.

- $J(\theta) = \mathbb{E}[r(x, y)]$
- $\nabla J(\theta) = \mathbb{E}[r(x, y) \nabla \log p_\theta(y \mid x)]$

$$= \mathbb{E}[r(x, y) \sum_{t=0}^{l} \nabla \log \pi_\theta(y_t \mid x, y_{0:t-1})]$$

$$p_\theta(y \mid x) = \prod_{t=0}^{l} \pi_\theta(y_t \mid x, y_{0:t-1})$$

RLHF for LLMs

Policy gradient is the gradient of a surrogate loss function:

$$L(\theta) = \mathbb{E}_{y \sim p_{\texttt{ref}}}[r(x, y) \log p_\theta(y \mid x)]$$

$$\nabla J(\theta) = \nabla L(\theta)$$

- If $p_{\texttt{ref}}$ is the current model's response distribution then we are doing on-policy policy gradient.
- If $p_{\texttt{ref}}$ is any other model, we are doing off-policy policy gradient.
- If $p_{\texttt{ref}}$ is responses generated by a human expert and $r(x, y) = 1$ then we are doing imitation learning (supervised finetuning).

Summary

- Imitation learning, inverse RL, and RLHF enable learning decision-making in RL environments by leveraging human knowledge.
- Often surpasses RL when we have access to an expert.
  - But potentially limits performance.
- Recent RLHF work enables humans to only specify preferences, which is often easier than absolute scores.
- RLHF enables training of LLMs to optimize for human objectives.