Reinforcement Learning from Human Feedback

Sanjiban Choudhury
The story so far ...

- Decision-making
- Perception
- Models of humans
Models of Humans

What humans want a robot to do?

What humans do around robots?
Let’s begin with Reinforcement Learning
We know how to make a RL block!

$R(s, a) \rightarrow \text{Your favorite RL algorithm} \rightarrow \pi^*(a | s)$
But how do we design reward function??

Your favorite
RL algorithm

\[ R(s, a) \rightarrow \pi^*(a | s) \]
Think-Pair-Share
Designing $R(s,a)$ for self-driving

Let’s say we wanted the robot to smoothly nudge around a parked car.
Think-Pair-Share!

Think (30 sec): What are the different components of the reward function you would code up? How would you assign weights to each component?

Pair: Find a partner

Share (45 sec): Partners exchange ideas
Some components of reward function

- Control Effort
- Proximity
- Boundary Violation
Manually tuning reward function to get the desired behavior is incredibly frustrating, time consuming, and does not scale.
Desiderata

1. Solve tasks where humans can recognize or demonstrate behavior

2. Allow agents to be taught by non-expert users

3. Scale to large problems

4. Economic with user feedback
What are better ways for humans to provide feedback to robots?
Think-Pair-Share
Think-Pair-Share!

Think (30 sec): What are the various ways for humans to provide feedback to the self-driving car?

Pair: Find a partner

Share (45 sec): Partners exchange ideas
Different types of feedback!

- Demonstrations
- Preference
- Interventions
- Ranking
- E-stops
- Language feedback
- Improvements
Let’s look at an example

Demonstrations

Preference

Interventions

Ranking

E-stops

Language feedback

Improvements
Recap: Learning to drive

Learnt policy

Demonstration

[SCB+ RSS’20]
Behavior Cloning crashes into a wall
What can’t we do DAGGER?
Problem: **Impractical** to query expert **everywhere**

Can we learn from **natural** human interaction, e.g., interventions?
Learn from natural human interventions?

Hands free, no corrections!
Learn from natural human interventions?

Take over and drive back!
EIL is "good-enough" after 60 sec of trials.
But ... we want a general solution that incorporates all feedback

- Demonstrations
- Preference
- Interventions
- Ranking
- E-stops
- Language feedback
- Improvements
Is there a way to **unify** feedback?

- Demonstrations
- Preference
- Interventions
- Ranking
- E-stops
- Language feedback
- Improvements
Is there a way to unify feedback?

- Demonstrations
- Interventions
  - Preference
- E-stops
- Ranking
- Language feedback
- Improvements

Reward Function

\[ R(s, a) \]
The simplest feedback:
Preferences
Deep Reinforcement Learning from Human Preferences

Paul F Christiano
OpenAI
paul@openai.com

Jan Leike
DeepMind
leike@google.com

Tom B Brown
nottombrown@gmail.com

Miljan Martic
DeepMind
miljanm@google.com

Shane Legg
DeepMind
legg@google.com

Dario Amodei
OpenAI
damodei@openai.com
Let’s work out the math!
How well does it perform on Reacher?

RL with learnt reward approaches RL with real rewards
How well does it perform on Ant?

RL with learnt reward approaches outperforms RL with real reward!

How?!
On the Ant task the human feedback significantly outperformed the synthetic feedback, apparently because we asked humans to prefer trajectories where the robot was “standing upright,” which proved to be useful reward shaping. (There was a similar bonus in the RL reward function to encourage the robot to remain upright, but the simple hand-crafted bonus was not as useful.)
Failure cases

On Qbert, our method fails to learn to beat the first level with real human feedback; this may be because short clips in Qbert can be confusing and difficult to evaluate.
Quiz
When can we perfectly recover the ground truth reward from preference?

When poll is active respond at PollEv.com/sc2582

Send sc2582 to 22333
How do we generalize Preferences to Ranking?
Let’s work out the math!
How do we generalize this idea to learning from interventions?
Learning Robot Objectives from Physical Human Interaction

Andrea Bajcsy*, Dylan P. Losey*, Marcia K. O'Malley, and Anca D. Dragan
How do we generalize this idea to learning from demonstrations?
Demonstrations are “preferred” trajectories

We can view demonstrations as positive trajectories.

But then where do we get negative trajectories from?

Key Idea: “Auto generate” negative trajectories by maximizing the current estimate of the reward
Inverse Reinforcement Learning

Apprenticeship Learning via Inverse Reinforcement Learning

Pieter Abbeel
Andrew Y. Ng
Computer Science Department, Stanford University, Stanford, CA 94305, USA
PABBEEL@CS.STANFORD.EDU
ANG@CS.STANFORD.EDU

Maximum Entropy Inverse Reinforcement Learning

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bziebart@cs.cmu.edu, anmaas@andrew.cmu.edu, dbagnell@ri.cmu.edu, anind@cs.cmu.edu

Generative Adversarial Imitation Learning

Jonathan Ho
Stanford University
hoj@cs.stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Of Moments and Matching:
A Game-Theoretic Framework for Closing the Imitation Gap

Gokul Swamy1 Sanjiban Choudhury2 J. Andrew Bagnell1,2 Zhiwei Steven Wu3
Begin with a guess of the reward function

$$R_{\theta}(\xi)$$
Optimize the current reward function to generate negative trajectories
$R_\theta(\xi)$

Push up

Push down

Animation from Sodhi et al. 2021
Animation from Sodhi et al. 2021
Animation from Sodhi et al. 2021

\[ R_\theta(\xi) \]

\[ \xi^h \]

\[ \xi \]
Animation from Sodhi et al. 2021
Gradients cancel

Animation from Sodhi et al. 2021
Inverse Reinforcement Learning as a Game

Do as well as the expert on any given reward function

$$\min_{\pi \in \Pi} \max_{R \in \mathcal{R}} J(\pi_E, R) - J(\pi, R)$$
Inverse Reinforcement Learning as a Game

Do as well as the expert on any given reward function

$$\min_{\pi \in \Pi} \max_{R \in \mathcal{R}} J(\pi_E, R) - J(\pi, R)$$

Reward player (No-Regret)  
Policy player (Best response)

$$R_i \leftarrow \arg \max_R \sum_j J(\pi_E, R) - J(\pi_j, R)$$  
$$\pi_{i+1} \leftarrow \arg \max_{\pi} J(\pi, R_i)$$

[Swamy et al.'21] Inverse Reinforcement Learning as a Game
Meta-algorithm for IRL

For $i = 1, \ldots, N$

Update reward estimate $R_i \leftarrow \arg \max \sum_j J(\pi_E, R) - J(\pi_j, R)$

(Bump up reward on expert, Bump down on learner)

Update policy $\pi_i \leftarrow \text{RL}(R_i)$

$\pi_{i+1} \leftarrow \arg \max \pi J(\pi, R_i)$