CSC415: Introduction to Reinforcement Learning

Lecture 9: Reinforcement Learning from Human Feedback

Dr. Amey Pore

Winter 2026

March 11, 2026

Guest speaker introduction

Dr. Mehdi Saeedi

- Principle Member of Technical Staff at AMD (Markham), since 2013
- Ph.D. (2010); postdoc at University of Southern California (USC)
- Over 25 granted patents; 50+ peer-reviewed publications in AI, intelligent agents, and compute systems
- At AMD: hardware and software in video games, inference acceleration, robotics
- Senior Member of IEEE; AMD AI/ML Patent Committee; Review Editor, TPC member, and track chair at leading conferences and journals

Lecture Outline

1. Human Feedback
2. Preference Learning
3. From Backflips to ChatGPT
4. Course Logistics

Human Feedback and Reinforcement Learning from Human Preferences

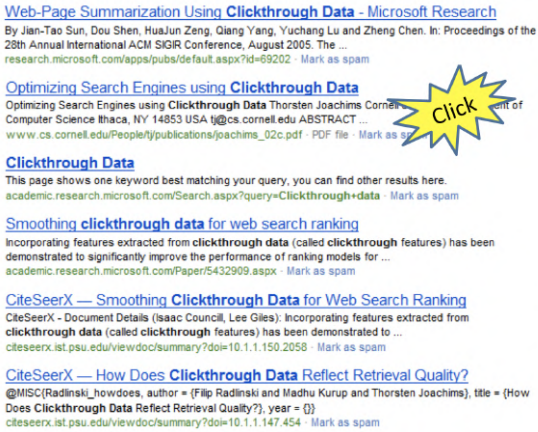
Human Input to Train RL Agents

- There are many ways for humans to help train RL agents
- This is relevant if we want RL agents that can match human performance and/or human values

Training a Robot Through Human and Environmental Feedback

[Thomaz et al., Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence 2008]

Comparing Recommendation Ranking Systems

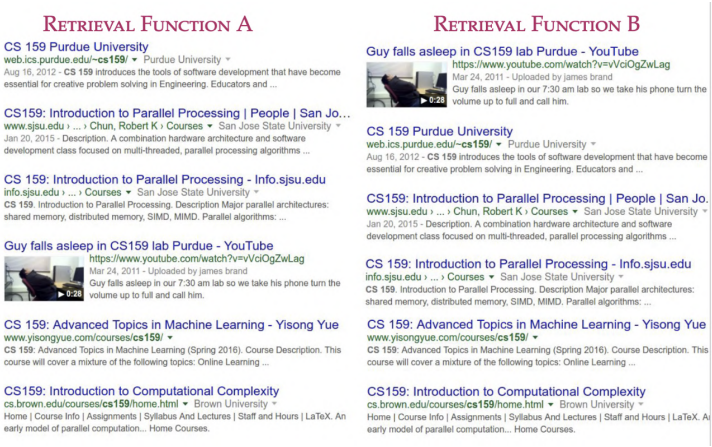


Interpretation 1:
Result #2 is good.
(Absolute)

Interpretation 2:
Result #2 is better
than Result #1.
(Relative / Preference)

[Yue, Slide from Yisong Yue. `https://sites.google.com/view/cs-159-spring2025/home`]

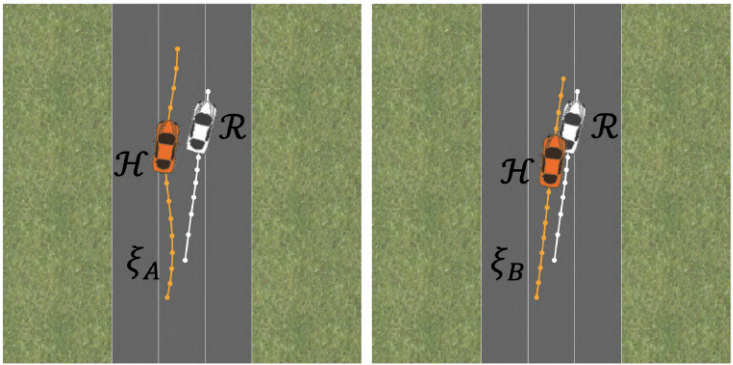
Comparing Recommendation Ranking Systems



[Yue, Slide from Yisong Yue. `https://sites.google.com/view/cs-159-spring2025/home`]

Active Learning of Preferences for Human Robot Interaction

$$\xi_A \text{ or } \xi_B \rightarrow I_t$$



[Sadigh et al., Active preference-based learning of reward functions. RSS 2017]

Pairwise Comparisons

- Often easier for people to make than hand writing a reward function
- Often easier than providing scalar reward (how much do you like this ad?)

Bradley-Terry Model (1952)

- Already saw with no other assumptions, the latent reward model is not unique
- Now focus on a particular structural model
- First consider simpler setting of k-armed bandits: K actions $b_1, b_2, \ldots b_k$. No state/context.
- Assume a human makes noisy pairwise comparisons, where the probability she prefers $b_i \succ b_j$ is

$$P(b_i \succ b_j) = \frac{\exp(r(b_i))}{\exp(r(b_i)) + \exp(r(b_j))} = p_{ij}$$

- Transitive: p_{ik} is determined from p_{ij} and p_{jk}

[Yue et al., The K-armed dueling bandits problem. JCSS 2012]

Definitions

Condorcet Winner

An item b_i is a **Condorcet winner** if for every other item b_j, $P(b_i \succ b_j) > 0.5$.

Copeland Winner

An item b_i is a **Copeland winner** if it has the highest number of pairwise victories against all other items. The score for an item is calculated as the number of items it beats minus the number of items it loses to.

Borda Winner

An item b_i is a **Borda winner** if it maximizes the expected score, where the score against item b_j is 1 if $b_i \succ b_j$ ($P(b_i \succ b_j) > 0.5$), 0.5 if $b_i = b_j$, and 0 if $b_i \prec b_j$.

Historically algorithms for k-armed or dueling (k=2) bandits focused on finding a Copeland winner.

Preference Learning

Fitting the Parameters of a Bradley-Terry Model

- First consider k-armed bandits: K actions $b_1, b_2, \ldots b_k$. No state/context.
- Assume a human makes noisy pairwise comparisons, where the probability she prefers $b_i \succ b_j$ is

$$P(b_i \succ b_j) = \frac{\exp(r(b_i))}{\exp(r(b_i)) + \exp(r(b_j))} = p_{ij}$$

Fitting the Parameters of a Bradley-Terry Model

- Assume a human makes noisy pairwise comparisons:

$$P(b_i \succ b_j) = \frac{\exp(r(b_i))}{\exp(r(b_i)) + \exp(r(b_j))} = p_{ij}$$

- Assume have N tuples of form (b_i, b_j, μ) where $\mu^{(1)} = 1$ if the human marked $b_i \succ b_j$, $\mu^{(1)} = 0.5$ if the human marked $b_i = b_j$, else 0 if $b_j \succ b_i$
- Maximize likelihood with cross entropy:

$$\mathcal{L} = -\sum_{(b_i, b_j, \mu) \in \mathcal{D}} \mu^{(1)} \log P(b_i \succ b_j) + (1 - \mu^{(1)}) \log P(b_j \succ b_i)$$

Preference to Reward Modeling for RL

- Can also do this for **trajectories**
- Consider two trajectories, $\tau^1(s_0, a_7, s_{14}, \ldots)$ and $\tau^2(s_0, a_6, s_{12}, \ldots)$
- Let $R^1 = \sum_{i=0}^{T-1} r_i^1$ be the (latent, unobserved) sum of rewards for trajectory τ^1 and similarly for R^2.
- Define the probability that a human prefers $\tau^1 \succ \tau^2$ as:

$$\hat{P}\left[\tau^1 \succ \tau^2\right] = \frac{\exp\left(\sum_{i=0}^{t-1} r_i^1\right)}{\exp\left(\sum_{i=0}^{t-1} r_i^1\right) + \exp\left(\sum_{i=0}^{t-1} r_i^2\right)}$$

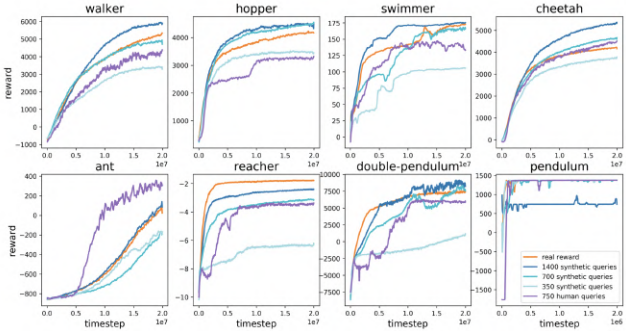
Preference to Reward Modeling for RL

- Define the probability that a human prefers $\tau^1 \succ \tau^2$ as:

$$\hat{P}\left[\tau^1 \succ \tau^2\right] = \frac{\exp\left(\sum_{i=0}^{t-1} r_i^1\right)}{\exp\left(\sum_{i=0}^{t-1} r_i^1\right) + \exp\left(\sum_{i=0}^{t-1} r_i^2\right)}$$

- Use **learned reward model**, and do **PPO** with this model

Reinforcement Learning from Human Feedback



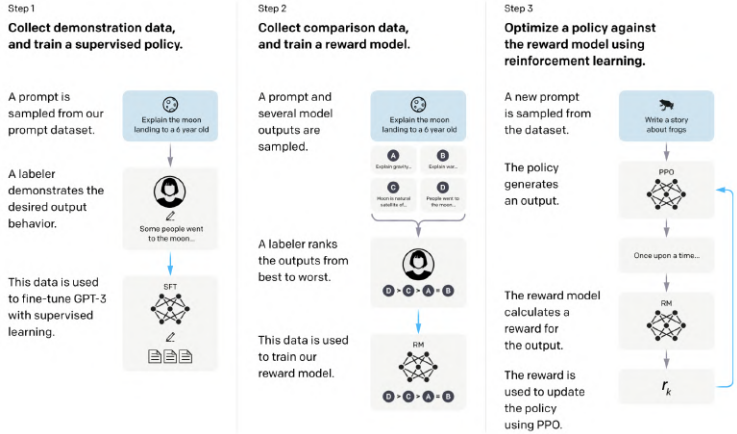
"needed 900 bits of feedback from a human evaluator to learn to backflip"

Learning to backflip: Human Feedback training process (Vimeo)

[Christiano et al., Deep RL from Human Preferences. NeurIPS 2017]

From Backflips to ChatGPT

High-level instantiation: 'RLHF' pipeline

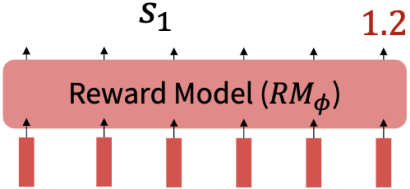


- First step: instruction tuning!
- Second + third steps: maximize reward (but how??)

How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

An earthquake hit San Francisco. There was minor property damage, but no injuries. > A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage. > The Bay Area has good weather but is prone to earthquakes and wildfires.



s_3 s_2

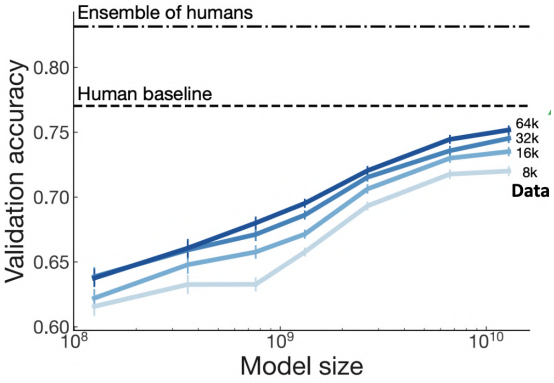
Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l)\sim D}\left[\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))\right]$$

"winning" "losing" s^w should score

Make sure your reward model works first!

- **Data**: Evaluate RM on predicting outcome of held-out human judgments
- Large enough RM trained on enough data approaching single human performance



[Stiennon et al., Learning to summarize with human feedback. NeurIPS 2020]

RLHF: Putting it all together

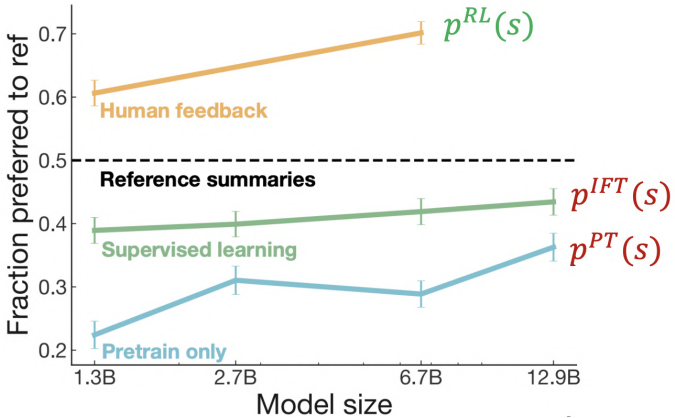
- We have everything we need:
  - A pretrained (possibly instruction-finetuned) LM $\pi_{\text{SFT}}(y)$
  - A reward model $r_\theta(y)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function
- Now to do RLHF:
  - Initialize a copy of the model $\pi_\phi^{RL}(y)$, with parameters ϕ we would like to optimize
  - Optimize the following reward with RL:

$$R(y) = r_\theta(y) - \beta \log \frac{\pi_\phi^{RL}(y)}{\pi_{\text{SFT}}(y)}$$

This is a penalty which prevents us from diverging too far from the pretrained model.
In expectation, it is the **KL divergence** between $\pi_\phi^{RL}(y)$ and $\pi_{\text{SFT}}(y)$.

[Christiano et al., Deep RL from Human Preferences. NeurIPS 2017; Stiennon et al., Learning to summarize with human

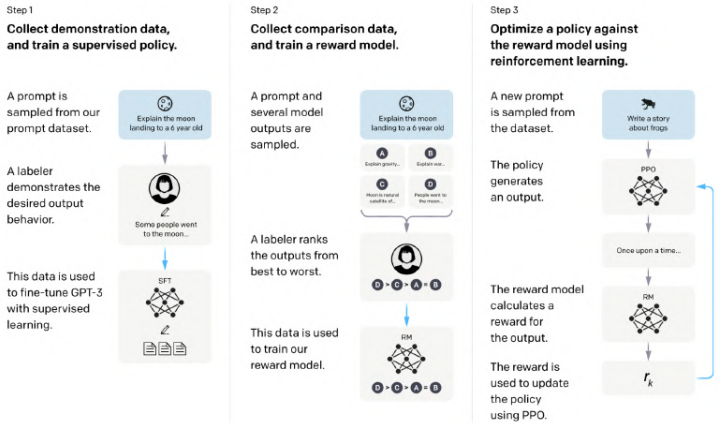
RLHF provides gains over pretraining + finetuning



[Stiennon et al., Learning to summarize with human feedback. NeurIPS 2020]

InstructGPT: scaling up RLHF to tens of thousands of tasks

30k



[Ouyang et al., Training language models to follow instructions with human feedback. NeurIPS 2022]

Controlled comparisons of "RLHF" style algorithms

- Many works study RLHF behaviors using **GPT-4 feedback** (Simulated) as a surrogate for human feedback
- **PPO** (method in InstructGPT) does work
- Simple baselines (**Best-of-n**, Training on 'good' outputs) works well too

[Dubois et al., AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. NeurIPS 2023]

Course Logistics

1. Assignment 1 grades will be released today.
2. Lab 3 grades will be released tomorrow.
3. Project proposal grades will be released over the weekend with feedback so you can work on it.
4. Quiz 2 will be on March 19th.
5. It will cover from lecture 5–8 (lecture 9 is optional).
6. Tomorrow's tutorial will be ungraded. It will cover aspects of statistical analysis, experimentation and writing.