An Introduction to RL from Human Feedback

1. Language as an MDP
2. Two Key Challenges of Fine-Tuning
3. Learning from Preferences
4. KL-Regularized RL
5. On the Information Geometry of RLHF

Gokul Swamy
gswamy@cmu.edu
www.gokul.dev

---

① $M = \{$

S: set of partial generations, e.g. all strings w/ len $\leq H$ → $|S| = |A|^H$

A: set of tokens, e.g. {'a', 'b', ...} → "byte-pair", watch Karpathy

T: $P(s' \mid s, a) = \begin{cases} 1, & s' = s \circ a \\ 0, & \text{o.w.} \end{cases}$ → "tree-structured", deterministic, known, resets easy

r: ???

ρ_0: prompts, e.g. "Summarize __".

H: maximum generation length.

$\}$

$\xi = [s_0, a_0, s_1, a_1, \ldots, s_H]$

= ["We", "The", "We The", "People", "We The People"]

"Next-Token Prediction"

2

LLM → Pre-training → Fine-Tuning → ChatGPT

Pre-training ↓ NTP on random internet text

Fine-Tuning:

1. "Supervised Fine-Tuning" (SFT): NTP on curated text $\Rightarrow \pi_{ref}$
2. RL from Human Feedback

Our focus for today

Challenge 1: What is r?

- the "reward design" problem

$\Rightarrow$ Leads to 3: learning from preferences

Challenge 2: How to stay "close" to π_{ref}?

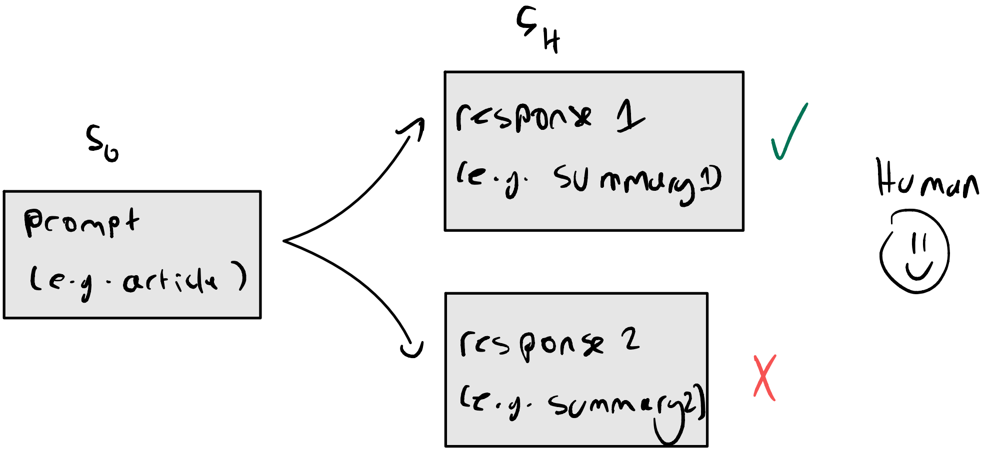
- the "fine-tuning" problem

$\Rightarrow$ Leads to 4: KL-regularized RL

③ <u>Reward Design</u> is hard for problems where the behavior we want is for an agent to be "human-like" (e.g. self-driving, chat-bots).

$\Rightarrow$ key idea: <u>learn</u> the reward function from data via MLE

$\Rightarrow$ for RLHF, we will learn from preference feedback



s_H' $\quad \Rightarrow D = \{(s_0, s_H^+, s_H^-)\}$

$\Rightarrow$ "Bradley-Terry" Model: $P_{r^*}(s_H^+ \succ s_H^-) = \dfrac{1}{1+\exp(r^*(s_H^-) - r^*(s_H^+))}$

$= \sigma(r^*(s_H^+) - r^*(s_H^-))$

"polite fiction", ask me later

$$\hat{r}_{BT} = \underset{r \in R}{\arg\min}\ D_{KL}(P_{r^*} \| P_r) \quad \text{(forward KL proj)}$$

$$= \underset{r \in R}{\arg\min}\ \mathbb{E}_{s_H^+, s_H^- \sim P_{r^*}}[\underbrace{\log P_{r^*}(s_H^+ \succ s_H^-)}_{\text{constant}} - \log P_r(s_H^+ \succ s_H^-)]$$

$$\approx \underset{r \in R}{\arg\min}\ \mathbb{E}_D[-\log P_r(s_H^+ \succ s_H^-)] \quad \text{(FKL = MLE)}$$

$$= \underset{r \in R}{\arg\min}\ \mathbb{E}_D[-\log \sigma(r(s_H^+) - r(s_H^-))]$$

4. Let us use π_{ref} to refer to the output of SFT. We want to stay "close" in policy space to π_{ref} during RLHF to not "exploit" the reward model. This leads to the <u>KL-Regularized RL Problem:</u>

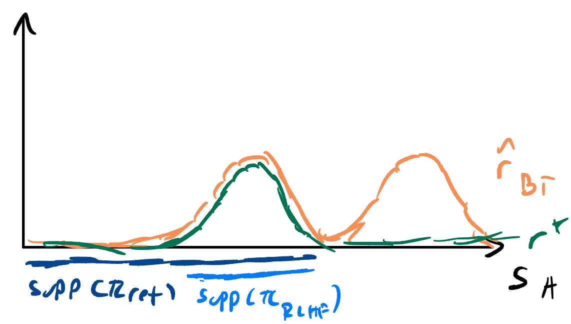
"reverse KL"

$$\boxed{\pi_{RLHF} = \underset{\pi \in \Pi}{\arg\max} \; \mathbb{E}_{\xi \sim \pi}\left[\hat{r}_{BT}(s_H)\right] - D_{KL}(\pi \,\|\, \pi_{ref})}$$

$$= \underset{\pi \in \Pi}{\arg\max} \; \mathbb{E}_{\xi \sim \pi}\left[\hat{r}_{BT}(s_H) - \sum_{h}^{H} \log \frac{\pi(a_h \mid s_h)}{\pi_{ref}(a_h \mid s_h)}\right]$$

↳ never pick a token ref. policy wouldn't

Solve using, e.g., PPO.



⇒ Intuitively, RLHF is doing "mode-selection" on top of SFT

Surprisingly enough, there is a closed form answer to the above KL-regularized RL problem at the trajectory level.

Let's work w/ trajectory-level distributions w/o prompts:

$$= \arg\max_{p} \mathbb{E}_{\xi \sim p} \left[\hat{r}_{BT}(\xi) - \log p(\xi) + \log p_{ref}(\xi) \right]$$

$$\text{s.t.} \quad \sum_{\xi} p(\xi) = 1.$$ => p is a valid prob. distribution

$$= \arg\max_{p} \min_{\lambda} \sum_{\xi} p(\xi) \left[\hat{r}_{BT}(\xi) - \log p(\xi) + \log p_{ref}(\xi) \right] + \lambda \left(\sum_{\xi} p(\xi) - 1 \right)$$

$$= \arg\max_{p} \min_{\lambda} \sum_{\xi} p(\xi) \left[\hat{r}_{BT}(\xi) - \log p(\xi) + \log p_{ref}(\xi) + \lambda \right] - \lambda$$

$$= \arg\max_{p} \min_{\lambda} L(p, \lambda)$$

Let's apply the stationarity condition from KKT: $\forall \xi$,

$$\nabla_{p(\xi)} L(p^*, \lambda^*) = 0$$

$$\Rightarrow \left[\hat{r}_{BT}(\xi) - \log p^*(\xi) + \log p_{ref}(\xi) + \lambda^* \right] + \cancelto{-1}{p^*(\xi) \left[\frac{-1}{p^*(\xi)} \right]} = 0$$

$$\Rightarrow \log p^*(\xi) = \hat{r}_{BT}(\xi) + \log p_{ref}(\xi) + \lambda^* - 1$$

$$\Rightarrow p^*(\xi) = \frac{p_{ref}(\xi) \cdot \exp(\hat{r}_{BT}(\xi))}{\exp(1 - \lambda)}$$

$$\nabla_{\lambda^*} L(p^*, \lambda^*) = \sum_{\xi} p^*(\xi) - 1 = 0$$

$$\Rightarrow \sum_{\xi} p_{ref}(\xi) \cdot \exp(\hat{r}_{BT}(\xi)) = \exp(1 - \lambda^*)$$

$$\Rightarrow \lambda^* = 1 - \log \left(\sum_{\xi} p_{ref}(\xi) \exp(\hat{r}_{BT}(\xi)) \right)$$

"exponential family" / "maximum entropy"

"partition function"

$$\boxed{\lambda^* = 1 - \log(Z), \quad p^*(\xi) = \frac{p_{ref}(\xi) \cdot \exp(\hat{r}_{BT}(\xi))}{Z}}$$

Now, consider the RKL projection of p^* onto some class $\mathbb{P}$:

$$\underset{p \in \mathbb{P}}{\arg\min} \; D_{KL}(p \,\|\, p^*)$$

$$= \underset{p \in \mathbb{P}}{\arg\min} \sum_{\xi} p(\xi) \left[\log p(\xi) - \log p^*(\xi) \right]$$

$$= \underset{p \in \mathbb{P}}{\arg\min} \sum_{\xi} p(\xi) \left[\log p(\xi) - \log p_{ref}(\xi) - \hat{r}_{BT}(s_H) + \overbrace{\log Z}^{\text{constant}} \right]$$

$$= \underset{p \in \mathbb{P}}{\arg\min} \sum_{\xi} p(\xi) \left[\log p(\xi) - \log p_{ref}(\xi) - \hat{r}_{BT}(s_H) \right]$$

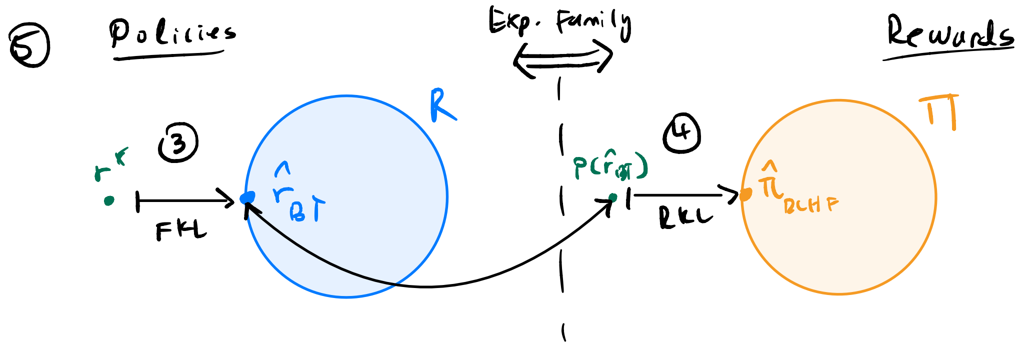
$$= \underset{p \in \mathbb{P}}{\arg\max} \sum_{\xi} p(\xi) \left[\hat{r}_{BT}(s_H) - \log \frac{p(\xi)}{p_{ref}(\xi)} \right]$$

$$= \underset{p \in \mathbb{P}}{\arg\max} \; \mathbb{E}_{\xi \sim p} \left[\hat{r}_{BT}(s_H) \right] + D_{KL}(p \,\|\, p_{ref})$$

Now, let's set $\mathbb{P} = \{ p(\xi) = \prod_h \pi(a_h \mid s_h) \mid \pi \in \Pi \}$ and swap:

$$= \underset{\pi \in \Pi}{\arg\max} \; \mathbb{E}_{\xi \sim \pi} \left[\hat{r}_{BT}(s_H) \right] + D_{KL}(\pi \,\|\, \pi_{ref})$$

$\Rightarrow$ Thus, Max Ent / soft RL is an RKL projection



Aside: We can easily incorporate contexts / prompts / init. states:

$$p^*(s_H \mid s_0) = \frac{\prod_h^H \pi_{ref}(a_h \mid s_h) \cdot \exp(\hat{r}_{BT}(s_H))}{Z(s_0)}$$

If you'd like to learn:

- how you actually solve the RL problem in practice
- why we can't just optimize the policy on pref. data
- how to do the above w/o Bradley Terry assumption

Take $\boxed{17-740}$ w/ Drew, Steven, and I next semester!

Website: <u>www.interactive-learning-algos.github.io</u>