Note on Pairwise Proximal Policy Optimization

Tianbing Xu

November 19, 2023

1 Introduction

For the Reinforcement Learning from Human Feedback(RLHF), we need a comparison of response sequences y_1 and y_2 for the same prompt sequence x either used to train a preference model explicitly ([3]) with PPO, or directly optimize the **Language Model** (or **Policy Network**) using DPO ([2]). The challenge lies in the difficulty of learning the absolute values of feedback rewards based on preference comparisons. Moreover, this approach may introduce instability due to unreliable reward estimation for the subsequent RL fine-tuning. A crucial insight is that, in this context, RL might benefit more from learning the relative feedbacks of rewards to enhance the training of language models. Consequently, akin to the approach in DPO, this paper ([1]) proposes the introduction of Pairwise Proximal Policy Optimization (P3O) tailored for handling relative feedbacks from pairwise preference comparison (Figure 1).

2 Method

The Language Generation Process can be conceptualized as a Context Bandit problem instead of a Markov Decision Process (MDP). In this context, given a prompt $\mathbf{x} = \{t_1, \ldots, t_h\}$, a sequence $\mathbf{y} = \{t_{h+1}, \ldots, t_n\}$ is generated by a Language Model $\pi_\theta(y|x)$. In the realm of Reinforcement Learning (RL), this model effectively functions as a policy network. For simplicity in the subsequent derivations, we omit the explicit representation of x and denote the model as $\pi_\theta(y)$. Eventually, the entire sequence (x, y) is evaluated using a reward function $\mathbf{r}(\mathbf{y})$.

2.1 Derivation of Pairwise Policy Gradient

We start from vanilla policy gradient (VPG),

$$\nabla L^{\mathrm{VPG}}(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[r(y)\nabla_\theta \log \pi_\theta(y)\right] = \sum_y \left[r(y)\nabla \pi_\theta(y)\right] \tag{1}$$

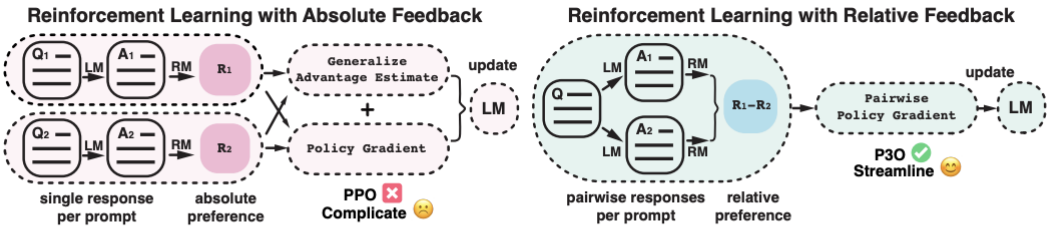

Figure 1: The figure on the left illustrates the prevalent method for fine-tuning LMs using RL, which relies on **Absolute Feedback**. In this paradigm, algorithms like PPO has to learn a V function, which capture not only the valuable relative preference information, but also less part, which is the scale of the reward for a given prompt. Contrastingly, the figure on the right presents paradigm for optimizing reward model trained via comparative loss, *e.g.*, Bradley-Terry Loss (Bradley & Terry, 1952). Our algorithm generate a pair of responses per prompt, leveraging only the **Relative Feedback** - derived from the difference in reward - for policy gradient updates. This method obviates the need for additional V function approximations and intricate components like GAE (Schulman et al., 2015b).

To reduce the variance, we plug in the base $b = \sum_y [r(y)\pi_\theta(y)]$,

$$\nabla L^{\text{VPG}}(\pi_\theta) = \sum_y [(r(y) - b)\nabla\pi_\theta(y)] = \sum_{y_1,y_2} [(r(y_1) - r(y_2))\pi_\theta(y_2)\nabla\pi_\theta(y_1)] \tag{2}$$

Define the delta of rewards as the relative feedback,

$$\delta_r(y_1, y_2; \theta) = r_\theta(y_1) - r_\theta(y_2) = \beta \left(\log \frac{\pi_\theta(y_1)}{\pi_0(y_1)} - \log \frac{\pi_\theta(y_2)}{\pi_0(y_2)} \right) \tag{3}$$

where, π_0 is the initial language model either from pre-trained or supervised fine-tuning, and the reward $r_\theta(y)$ (omitting x for notation simplicity) is derived from the KL-constrained optimal policy,

$$r_\theta(y|x) = \beta \left(\log \frac{\pi_\theta(y|x)}{\pi_0(y|x)} \right) + \beta \log Z(x) \tag{4}$$

Swap y_1 and y_2, we arrive at Pairwise Policy Gradient (PPG),

$$\begin{aligned} \nabla L^{\text{PPG}}(\pi_\theta) &= \mathbb{E}_{y_1,y_2\sim\pi_\theta} \left[(r_\theta(y_1) - r_\theta(y_2))\nabla_\theta(\log \pi_\theta(y_1) - \log \pi_\theta(y_2)) \right] \\ &= \frac{1}{\beta} \mathbb{E}_{\pi_\theta} \left[\delta_r(\mathbf{y}_1, \mathbf{y}_2; \theta)\nabla_\theta \delta_r(\mathbf{y}_1, \mathbf{y}_2; \theta) \right] \end{aligned} \tag{5}$$

Interestingly, the policy gradient comprises two components: one is the relative feedback of rewards, and the other is the gradient of the relative feedback with respect to the policy network (or language model) parameter θ. It is clear that the policy gradient increases in the direction of relative feedback.

2.2 Compared to DPO

The cross-entropy loss of Direct Preference Optimization w.r.t the policy (omitting x for formula simplicity),

$$\begin{aligned} L^{\text{DPO}}(\pi_\theta) &= -\mathbb{E}_D \left[\log P(y_1 > y_2|x) \right] = -\mathbb{E}_D \left[\log \sigma \left(\delta_r(y_1, y_2; \theta) \right) \right] \\ &= -\mathbb{E}_D \left[\log \sigma \left(\beta \log \left(\frac{\pi_\theta(y_2)}{\pi_0(y_2)} \right) - \beta \log \left(\frac{\pi_\theta(y_1)}{\pi_0(y_1)} \right) \right) \right] \end{aligned} \tag{6}$$

where $\hat{r}_\theta(y) = \beta \left(\log \pi_\theta(y) - \log \pi_0(y) \right)$ is the proxy reward, and $\delta_r(y_1, y_2; \theta) = \hat{r}_\theta(y_1) - \hat{r}_\theta(y_2)$ is the relative feedback reward without the need to consider the normalization term $Z(x)$ in the absolute feedback reward (see Eq.4).

Then, the policy gradient (DPO) is (positive for max likelihood),

$$\begin{aligned} \nabla_\theta L^{\text{DPO}}(\pi_\theta) &= \beta \mathbb{E}_D \left[\sigma(r_\theta(y_2) - r_\theta(y_1)) \left(\nabla_\theta \log \pi_\theta(y_1) - \nabla_\theta \log \pi_\theta(y_2) \right) \right] \\ &= \mathbb{E}_D \left[\sigma(\delta_r(\mathbf{y}_2, \mathbf{y}_1; \theta))\nabla_\theta \delta_r(\mathbf{y}_1, \mathbf{y}_2; \theta) \right] \end{aligned} \tag{7}$$

Compared to Pairwise Policy Gradient, with the same gradient of the relative feedback (and ignoring the temperature β), DPO increases in the direction of the sigmoid smoothing function of relative feedback. Furthermore, the response sequences of feedback are sampled from the offline policy (π_0) or a fixed distribution instead of the online policy distribution (π_θ).

2.3 Compared to PPO

We optimize the policy network(or language model) to maximize the expected rewards (omitting x for formula simplicity),

$$max_{\pi_\theta} \mathbb{E}_{\pi_\theta} [r_\phi(y) - V_\phi(x)] - \beta D_{KL} \left[\pi_\theta(y) || \pi_0(y) \right] \tag{8}$$

where $V_\phi(x) = \mathbb{E} \left[r_\phi(y|x) \right]$ is a value function introduced to reduce the variance of policy gradient.

In contrast to P3O and DPO, this is the absolute feedback reward.

$$r(y) = r_\phi(y) - V_\phi(x) - \beta \left(\log \pi_\theta(y) - \log \pi_0(y) \right) \tag{9}$$

References

[1] Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran and Jiantao Jiao, University of California, Berkeley *Pairwise Proximal Policy Optimization: Harnessing Relative Feedback for LLM Alignment, 2023*

[2] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, Chelsea Finn, *Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2023*

[3] L. Ouyang, et. al OpenAI, *Training language models to follow instructions with human feedback, NIPS 2022*