Large Language Models

Direct Preference Optimization

Mohammad Hossein Rohban

Fall 2023

Courtesy: Most of the slides are adopted from the papers by R. Rafailov et al. 2023 “Direct Preference Optimization: Your Language Model is Secretly a Reward Model”

Motivation

- RLHF is a complex and unstable process.
  - A lot of knobs such as β, controlling the KL divergence term.
- Can we directly optimize the preference function?
  - Represented by the LLM itself.

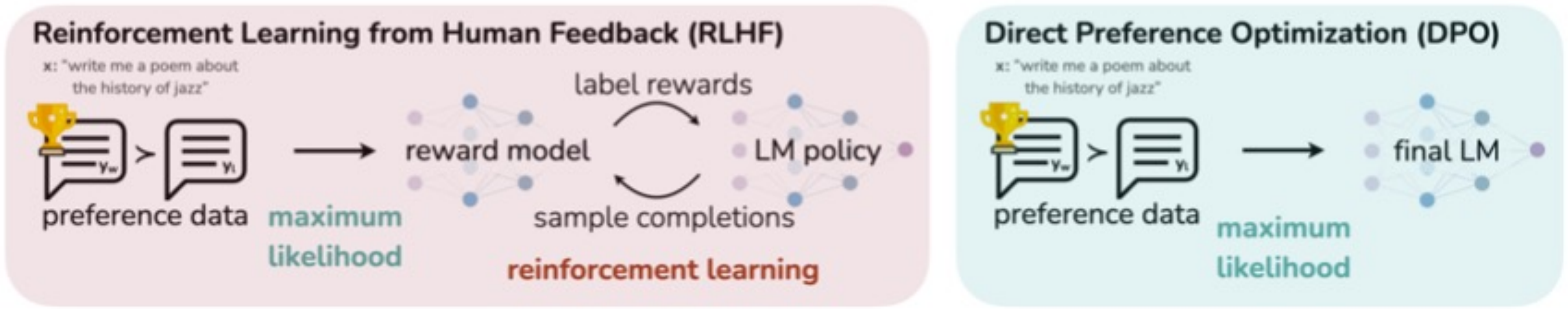

Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, without an explicit reward function or RL.

How to go about it?

- First, note that the optimal solution to this RL problem

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right] - \beta\mathbb{D}_{\textbf{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\text{ref}}(y \mid x)\right]$$

is $$\pi_r(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x, y)\right).$$

- How? Form the Lagrangian:
- $$\mathcal{L} = \int \left(r_\phi(x, y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}\right)\pi_\theta(y|x)p(x)dxdy + \lambda\left(1 - \int \pi_\theta(y|x)p(x)dxdy\right).$$

How to go about it? (cont.)

- Now, for any particular value of (x, y), take the derivative of the Lagrangian w.r.t. $\pi_\theta(y|x)$ and find its roots:
- $\frac{\partial \mathcal{L}}{\partial \pi_\theta(y|x)} = \left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}\right) p(x) - \beta p(x) - \lambda p(x) = 0.$
- $\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} = \exp\left\{\frac{1}{\beta} r_\phi(x,y)\right\} . \exp\left\{-\frac{\lambda+\beta}{\beta}\right\}$
- $\pi_\theta(y|x) = \exp\left\{-\frac{\lambda+\beta}{\beta}\right\} \pi_{ref}(y|x) \exp\left\{\frac{1}{\beta} r_\phi(x,y)\right\}$

Can we decipher the reward function from π?

- Solving $\pi_\theta(y|x) = \underbrace{\exp\left\{-\frac{\lambda+\beta}{\beta}\right\}}_{1/Z(x)} \pi_{ref}(y|x)\ \exp\left\{\frac{1}{\beta} r_\phi(x,y)\right\}$ for the r.

- Therefore, $r_\phi(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$.

Now apply the loss for learning reward!

- Recall: $\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$
- Now replace $r_\phi(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$ into this Eq.
- It becomes:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

How does the grad. update look like?

- Recall that:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

- Therefore:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$-\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \Bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \Bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l} \Bigg] \Bigg]$$

$$\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

How to interpret this?

- It's a weighted next token predictor loss.
- It gets larger weight whenever the relative ordering of the winner and loser completions are not correct.

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \Bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Bigg]\Bigg]$$

Tasks

- **Positive sentiment generation**: Given prefix of a movie review from IMDb dataset, y is the completion with positive sentiment.
- **Summarization**: Summarize a given forum post from Reddit; the TL;DR Reddit dataset.

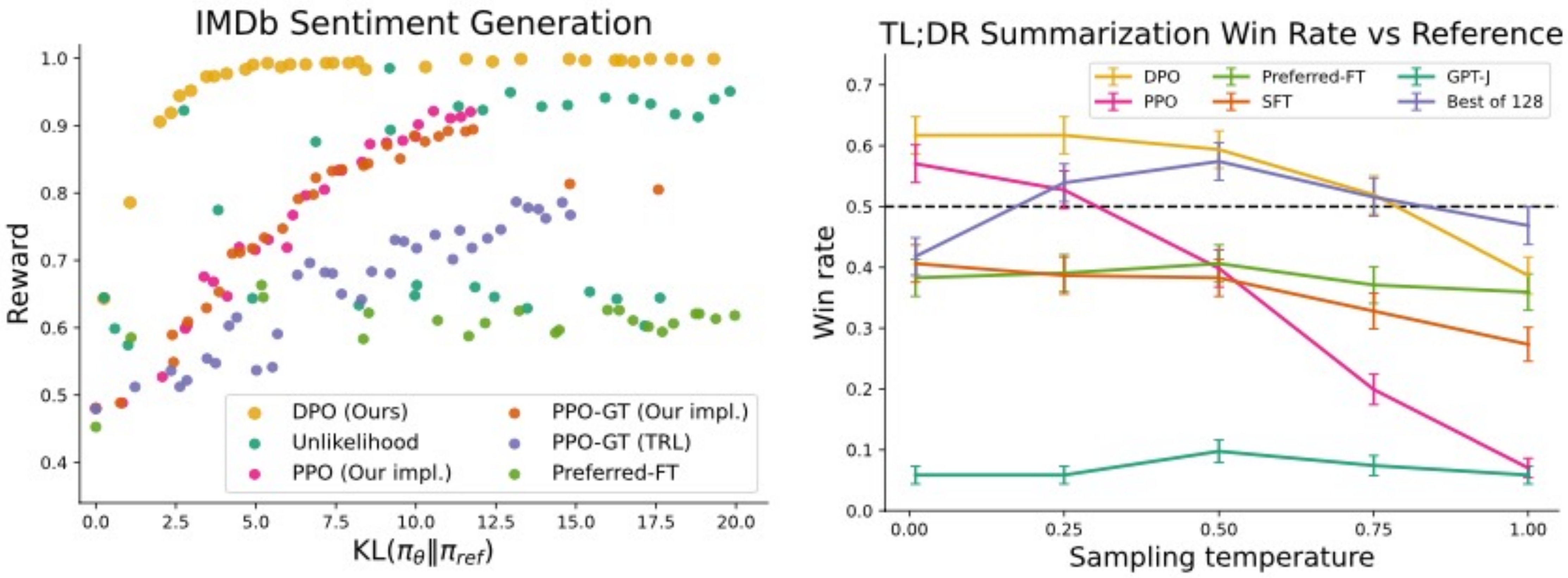

Figure 2: **Left.** The frontier of expected reward vs KL to the reference policy. DPO provides the highest expected reward for all KL values, demonstrating the quality of the optimization. **Right.** TL;DR summarization win rates vs. human-written summaries, using GPT-4 as evaluator. DPO exceeds PPO's best-case performance on summarization, while being more robust to changes in the sampling temperature.