language model alignment: story and math

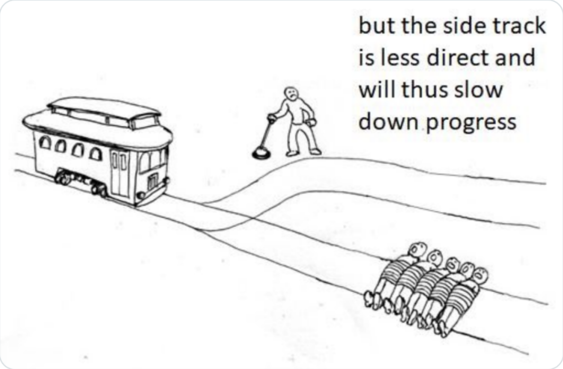
Ruizhe Shi

Institute for Interdisciplinary Information Sciences, Tsinghua University

Nov. 17, 2024

Contents

Value alignment



To align the value or intention of AI system with humanity.

Ability alignment

Prompt: In Bash, how do I list all text files in the current directory (excluding subdirectories) that have been modified in the last month

Response: You can use the 'find' command in Bash to list all text files in the current directory (excluding subdirectories) that have been modified in the last month. The command would be:

```
find . -maxdepth 1 -type f -mtime -30 -name "*.txt"
```

Here's a breakdown of the command:
* '.' specifies the current directory.
* '-maxdepth 1' tells 'find' to only search the current directory and not any subdirectories.
* '-type f' tells 'find' to only search for files (not directories or other types of files).
* '-mtime -30' tells 'find' to only search for files that have been modified in the last 30 days.
* '-name "*.txt"' tells 'find' to only search for files with the '.txt' extension.
Note that the '-mtime' option uses the number of days since the file was last modified, so '-mtime -30' means "modified in the last 30 days". If you want to search for files modified in the last month, you can use the '-mtime -30' option.

Figure 1: Example of response of CODE LLAMA - INSTRUCT (34B) when queried for a specific shell command.

To enhance the ability of AI in certain areas (coding/math).

Style alignment

魔幻现实主义

多年以后，面对食死徒，哈利·波特将会回想起海格说他是巫师的那个遥远的夜晚。

古典文学

鲁伯入，曰：“哈利，汝乃巫师也。”

轻小说

呐，哈酱。虽然有点晚了，但我还是来了。抱歉呵，已经不用再害怕了，因为你不是一个人了哦。现在我要把真相告诉你，呐，你是一名巫师哟~~

To transfer the writing style of generated contents.

Preliminary

Notations

Each prompt is a state $x \in \mathcal{X}$.

Preliminary

Notations

Each prompt is a state $x \in \mathcal{X}$.

Each response is an action $y \in \mathcal{Y}$.

Preliminary

Notations

Each prompt is a state $x \in \mathcal{X}$.
Each response is an action $y \in \mathcal{Y}$.
The reward function is a mapping $R : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.

Preliminary

Notations

Each prompt is a state $x \in \mathcal{X}$.
Each response is an action $y \in \mathcal{Y}$.
The reward function is a mapping $R : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
Each policy is a mapping $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$.

Preliminary

Notations

Each prompt is a state $x \in \mathcal{X}$.
Each response is an action $y \in \mathcal{Y}$.
The reward function is a mapping $R : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
Each policy is a mapping $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$.

Preference

A preference model $p^*(y_1 \succ y_2 | x)$ indicates the probability that y_1 is preferred over y_2 given x by the annotator (human).
Given (x, y_1, y_2), we observe a sample $p \sim \text{Bernoulli}(p^*(y_1 \succ y_2 | x))$,

RLHF

Preference dataset

We assume the existence of a human preference dataset, $\mathcal{D} = \{(x^{(i)}, {y_w}^{(i)}, {y_l}^{(i)}\}_{i=1}^{N}$. Here the preference is observed as ${y_w}^{(i)} \succ {y_l}^{(i)}$.

RLHF

Preference dataset

We assume the existence of a human preference dataset, $\mathcal{D} = \{(x^{(i)}, {y_w}^{(i)}, {y_l}^{(i)}\}_{i=1}^{N}$. Here the preference is observed as ${y_w}^{(i)} \succ {y_l}^{(i)}$.

Reward learning

$$\mathcal{L}_r(\phi) = -\frac{1}{N}\sum_{i=1}^{N} \log \sigma(r_\phi(x^{(i)}, {y_w}^{(i)}) - r_\phi(x^{(i)}, {y_l}^{(i)}))\ ,$$

which is implicitly a sum of cross entropy loss.

RLHF

Preference dataset

We assume the existence of a human preference dataset, $\mathcal{D} = \{(x^{(i)}, {y_w}^{(i)}, {y_l}^{(i)}\}_{i=1}^{N}$. Here the preference is observed as ${y_w}^{(i)} \succ {y_l}^{(i)}$.

Reward learning

$$\mathcal{L}_r(\phi) = -\frac{1}{N}\sum_{i=1}^{N} \log \sigma(r_\phi(x^{(i)}, {y_w}^{(i)}) - r_\phi(x^{(i)}, {y_l}^{(i)}))\ ,$$

which is implicitly a sum of cross entropy loss.

Policy learning

PPO.

DPO

Closed-form solution

With no restrictions on parameterization, the optimal policy is

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x) \exp(r(x, y)/\beta) \ .$$

DPO

Closed-form solution

With no restrictions on parameterization, the optimal policy is

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)\ .$$

Policy learning

$$\mathcal{L}_\pi(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log\sigma\left(\beta\log\frac{\pi_\theta({y_w}^{(i)}|x^{(i)})}{\pi_{\text{ref}}({y_w}^{(i)}|x^{(i)})} - \beta\log\frac{\pi_\theta({y_l}^{(i)}|x^{(i)})}{\pi_{\text{ref}}({y_l}^{(i)}|x^{(i)})}\right)\ .$$

Starting with a question

Policy mixing

Give you two policies π_1, π_2, and $w_1, w_2 \in \mathbb{R}$, then how to efficiently decode a response y to maximize $\pi_1^{w_1}(y|x)\pi_2^{w_2}(y|x)$ with an acceptable error?

A common approach

Parameter merging (it is widely used)

Suppose π_1, π_2 are parameterized by θ_1, θ_2 respectively. then parameter merging is to produce a new policy π' parameterized by $w_1\theta_1 + w_2\theta_2$.

A common approach

Parameter merging (it is widely used)

Suppose π_1, π_2 are parameterized by θ_1, θ_2 respectively. then parameter merging is to produce a new policy π' parameterized by $w_1\theta_1 + w_2\theta_2$.

This approach clearly has no theoretical guarantee, especially when faced with complicated non-linear architecture like **Transformer**.

A common approach

Parameter merging (it is widely used)

Suppose π_1, π_2 are parameterized by θ_1, θ_2 respectively. then parameter merging is to produce a new policy π' parameterized by $w_1\theta_1 + w_2\theta_2$.

This approach clearly has no theoretical guarantee, especially when faced with complicated non-linear architecture like **Transformer**.

Here is an easy way to hack it: suppose that we have a policy π_a, which is built up with self-attention blocks, now we reverse the sign of Q, K matrices in its last block ($Q^\top K$ will not change), and get π_b. To get $\pi_a^{0.5}\pi_b^{0.5}$, can we directly merge their parameters? Clearly no.

Another approach

Greedy decoding

There is a natural assumption in NLP: given a policy π and a prompt x, we can greedily decode a response y to maximize $\pi(y|x)$ with an acceptable error. The approach is to autoregressively let

$$y_t = \arg\max_s \pi(s|x, y_{<t}) ,$$

where y_t is the t_{th} token of y.

Inspired by greedy decoding, here we provide a more theoretically-sound approach. Logit is defined as $z_k(y_t|x, y_{<t}) := \log \pi_k(y_t|x, y_{<t}) + \text{offset}(x, y_{<t})$. Let $z^\star := w_1 z_1 + w_2 z_2$. Then we have

Inspired by greedy decoding, here we provide a more theoretically-sound approach. Logit is defined as $z_k(y_t|x, y_{<t}) := \log \pi_k(y_t|x, y_{<t}) + \text{offset}(x, y_{<t})$. Let $z^\star := w_1 z_1 + w_2 z_2$. Then we have

$$
\begin{aligned}
&\arg\max_{y\in\mathcal{Y}} \pi_1^{w_1}(y|x)\pi_2^{w_2}(y|x) \\
&= \arg\max_{y\in\mathcal{Y}} w_1 \log \pi_1(y|x) + w_2 \log \pi_2(y|x) \\
&= \arg\max_{y\in\mathcal{Y}} \sum_{t=0}^{|y|} w_1 z_1(y_t|x, y_{<t}) + w_2 z_2(y_t|x, y_{<t}) \\
&= \arg\max_{y\in\mathcal{Y}} \underbrace{\sum_{t=0}^{|y|} z^\star(y_t|x, y_{<t})}_{\text{we can do greedy decoding here!}}
\end{aligned}
$$

Logit mixing

Logit mixing

Give you two policies π_1, π_2, and $w_1, w_2 \in \mathbb{R}$, then we can autoregressively let

$$y_t := \arg\max_s \pi_1^{w_1}(s|x, y_{<t})\pi_2^{w_2}(s|x, y_{<t}) \ ,$$

to maximize $\pi_1^{w_1}(y|x)\pi_2^{w_2}(y|x)$ with an acceptable error.

Application of logit mixing

What can we do with such an efficient policy mixing approach?

Application of logit mixing

What can we do with such an efficient policy mixing approach?

Recall RLHF & DPO

With no restrictions on parameterization, the optimal policy is

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta) \; .$$

Application of logit mixing

What can we do with such an efficient policy mixing approach?

Recall RLHF & DPO

With no restrictions on parameterization, the optimal policy is

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)\ .$$

Multi-objective alignment

Given $\pi_1 \propto \pi_{\text{ref}}(y|x)\exp(r_1(x,y)/\beta)$, $\pi_2 \propto \pi_{\text{ref}}(y|x)\exp(r_2(x,y)/\beta)$, where r_1, r_2 are two different reward functions, then we can efficiently obtain

$$\pi^\star(y|x) \propto \pi_{\text{ref}}(y|x)\exp((w_1 r_1(x,y) + w_2 r_2(x,y))/\beta)\ ,$$

for any $w_1, w_2 \in \mathbb{R}$. *For example, given a helpful agent and a harmless agent, we can flexibly balance them, without retraining a new model.*

Application of logit mixing

What can we do with such an efficient policy mixing approach?

Hyper-parameter adjustment

Given $\pi \propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)$, $\beta' \in \mathbb{R}_+$, then we can efficiently obtain

$$\begin{aligned}\pi^\star(y|x) &\propto \pi_{\text{ref}}^{1-\beta/\beta'}(y|x)\pi^{\beta/\beta'}(y|x)\\ &\propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta')\ .\end{aligned}$$

Thus we can efficiently adjust the RLHF hyper-parameter β without restarting training.

Application of logit mixing

What can we do with such an efficient policy mixing approach?

Proxy tuning

Given $\pi \propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)$, π', then we can efficiently obtain

$$\begin{aligned}\pi^\star(y|x) &\propto \pi'(y|x)\frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} \\ &\propto \pi'(y|x)\exp(r(x,y)/\beta)\ .\end{aligned}$$

Thus we can first conduct RLHF on a small proxy model (like 7B), then plug it into a large model (like 70B) and achieve equivalent effect.

Application of logit mixing

What can we do with such an efficient policy mixing approach?

Jail breaking

Given $\pi \propto \pi_{\text{ref}}(y|x)\exp(r(x,y)/\beta)$, π', then we can efficiently obtain

$$\begin{aligned}\pi^\star(y|x) &\propto \pi'(y|x)\frac{\pi_{\text{ref}}(y|x)}{\pi(y|x)} \\ &\propto \pi'(y|x)\exp(-r(x,y)/\beta)\ .\end{aligned}$$

Thus we can first conduct RLHF on a small proxy model (like 7B), then hack a strong model to make it unsafe or privacy-leaking.

Rethinking DPO

DPO policy learning

$$\mathcal{L}_\pi(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log \sigma\left(\beta \log \frac{\pi_\theta(y_w{}^{(i)}|x^{(i)})}{\pi_{\text{ref}}(y_w{}^{(i)}|x^{(i)})} - \beta \log \frac{\pi_\theta(y_l{}^{(i)}|x^{(i)})}{\pi_{\text{ref}}(y_l{}^{(i)}|x^{(i)})}\right) .$$

Rethinking DPO

DPO policy learning

$$\mathcal{L}_\pi(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log\sigma\left(\beta\log\frac{\pi_\theta({y_w}^{(i)}|x^{(i)})}{\pi_{\text{ref}}({y_w}^{(i)}|x^{(i)})} - \beta\log\frac{\pi_\theta({y_l}^{(i)}|x^{(i)})}{\pi_{\text{ref}}({y_l}^{(i)}|x^{(i)})}\right).$$

Does this ensure that $\pi_\theta(y_w)\uparrow$, $\pi_\theta(y_l)\downarrow$?

Preparation

Let $\hat{r}_\theta(x, y)$ denote $\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$. For a single sample (x, y_w, y_l),

$$\begin{aligned}
\mathcal{L}_\pi(\theta) &= -\log \sigma\left(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)\right) ; \\
\nabla_\theta \mathcal{L}_\pi(\theta) &= -\beta \frac{\sigma'\left(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)\right)}{\sigma\left(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)\right)} \left(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)\right) \\
&= -\beta \sigma\left(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)\right)\left(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)\right) .
\end{aligned}$$

Preparation

Let $\hat{r}_\theta(x, y)$ denote $\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$. For a single sample (x, y_w, y_l),

$$\mathcal{L}_\pi(\theta) = -\log \sigma(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)) \; ;$$

$$\begin{aligned}\nabla_\theta \mathcal{L}_\pi(\theta) &= -\beta \frac{\sigma'(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))}{\sigma(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))} (\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)) \\ &= -\beta \sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)) \; .\end{aligned}$$

One-step gradient descent

Denote $c(\theta) := \eta\beta\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))$, then

$$\Delta\theta = c(\theta)(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)) \; .$$

One-step gradient descent

Denote $c(\theta) := \eta\beta\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))$, then

$$\Delta\theta = c(\theta)(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)) \ .$$

By first-order Taylor-expansion, we have

One-step gradient descent

Denote $c(\theta) := \eta\beta\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))$, then

$$\Delta\theta = c(\theta)(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)) \ .$$

By first-order Taylor-expansion, we have

$$\begin{aligned}
\Delta \log \pi_\theta(y_w|x) &= \log \pi_{\theta+\Delta\theta}(y_w|x) - \log \pi_\theta(y_w|x) \\
&\approx \Delta\theta \cdot \nabla_\theta \log \pi_\theta(y_w|x) \\
&= c(\theta)(\|\nabla_\theta \log \pi_\theta(y_w|x)\|^2 - \langle \nabla_\theta \log \pi_\theta(y_w|x), \nabla_\theta \log \pi_\theta(y_l|x)\rangle) \ , \\
\Delta \log \pi_\theta(y_l|x) &= \log \pi_{\theta+\Delta\theta}(y_l|x) - \log \pi_\theta(y_l|x) \\
&\approx \Delta\theta \cdot \nabla_\theta \log \pi_\theta(y_l|x) \\
&= c(\theta)(\langle \nabla_\theta \log \pi_\theta(y_w|x), \nabla_\theta \log \pi_\theta(y_l|x)\rangle - (\|\nabla_\theta \log \pi_\theta(y_l|x)\|^2) \ .
\end{aligned}$$

Gradient Entanglement

The chosen log-probability change $\nabla \log \pi(y_w|x)$ depends on the rejected gradient $\nabla \log \pi(y_l|x)$, and similarly, the rejected log-probability $\Delta \log(y_l|x)$ change depends on the chosen gradient $(y_l|x)$.

$$\Delta \log \pi_\theta(y_w|x) \approx c(\theta)(\|\nabla_\theta \log \pi_\theta(y_w|x)\|^2 - \langle \nabla_\theta \log \pi_\theta(y_w|x), \nabla_\theta \log \pi_\theta(y_l|x)\rangle) \; ,$$

$$\Delta \log \pi_\theta(y_l|x) \approx c(\theta)(\langle \nabla_\theta \log \pi_\theta(y_w|x), \nabla_\theta \log \pi_\theta(y_l|x)\rangle - (\|\nabla_\theta \log \pi_\theta(y_l|x)\|^2) \; .$$

Gradient entanglement

$\Delta \log \pi_w, \Delta \log \pi_l$	$\log \pi_w, \log \pi_l$	**Condition**
$\Delta \log \pi_w \geq 0 \geq \Delta \log \pi_l$	$\log \pi_w \uparrow \log \pi_l \downarrow$	$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_w\|^2, \|\nabla \log \pi_l\|^2$
$0 \geq \Delta \log \pi_w \geq \Delta \log \pi_l$	$\log \pi_w \downarrow \log \pi_l \downarrow$	$\|\nabla \log \pi_w\|^2 \leq \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_l\|^2$
$\Delta \log \pi_w \geq \Delta \log \pi_l \geq 0$	$\log \pi_w \uparrow \log \pi_l \uparrow$	$\|\nabla \log \pi_l\|^2 \leq \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_w\|^2$

Table: Three possible cases of the changes on chosen and rejected log-probabilities in DPO. $\uparrow$ and $\downarrow$ indicate increase and decrease. **Case 1 (Ideal)**: $\log \pi_w$ increases and $\log \pi_l$ decreases; **Case 2**: $\log \pi_w$ and $\log \pi_l$ both decreases but $\log \pi_l$ decreases more; **Case 3**: $\log \pi_w$ and $\log \pi_l$ both increases but $\log \pi_w$ increases more.

One possible solution and remaining questions

Pairwise normalized gradient descent

We can modify the gradient update rule for the DPO loss as

$$\nabla_\theta \mathcal{L}_\pi(\theta) = -\beta\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)) \left(\frac{\nabla_\theta \log \pi_\theta(y_w|x)}{\|\nabla_\theta \log \pi_\theta(y_w|x)\|} - \frac{\nabla_\theta \log \pi_\theta(y_l|x)}{\|\nabla_\theta \log \pi_\theta(y_w|x)\|} \right) .$$

One possible solution and remaining questions

Pairwise normalized gradient descent

We can modify the gradient update rule for the DPO loss as

$$\nabla_\theta \mathcal{L}_\pi(\theta) = -\beta\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)) \left(\frac{\nabla_\theta \log \pi_\theta(y_w|x)}{\|\nabla_\theta \log \pi_\theta(y_w|x)\|} - \frac{\nabla_\theta \log \pi_\theta(y_l|x)}{\|\nabla_\theta \log \pi_\theta(y_w|x)\|} \right) .$$

If so, the gradient entanglement issue can be naturally solved, but

- How can this be practical?
- Do we really need to increase the π_w and decrease the π_l? Why?

These questions might be interesting and worth exploring.

More than reward

We already know that Your language model is secretly a reward function.

More than reward

We already know that Your language model is secretly a reward function.

But they are still different: the reward model only produces one signal for a complete response y, while the policy model produces signals for each token y_t, *i.e.*

More than reward

We already know that Your language model is secretly a reward function.

But they are still different: the reward model only produces one signal for a complete response y, while the policy model produces signals for each token y_t, *i.e.*

$$r(x, y) \text{ v.s. } \pi(y_1|x),\ \pi(y_2|x, y_0), \ldots,\ \pi(y_t|x, y_{1,\ldots,t-1})\ .$$

It reminds us what?

More than reward

We already know that Your language model is secretly a reward function.

But they are still different: the reward model only produces one signal for a complete response y, while the policy model produces signals for each token y_t, *i.e.*

$$r(x, y) \text{ v.s. } \pi(y_1|x),\ \pi(y_2|x, y_0), \ldots,\ \pi(y_t|x, y_{1,\ldots,t-1})\ .$$

It reminds us what?

Think about the per-step stuff related to reward:

- Value function $V(s)$.
- Q-function $Q(s, a)$.
- Advantage function $A(s, a) = Q(s, a) - V(s)$.

Preliminary

Max-entropy RL (Soft RL)

Suppose we have per-token reward $r(s_t, a_t)$, initial state distribution $\rho(s_0)$, then the KL-constrained RL objective is

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{a_t \sim \pi_\theta(\cdot|s_t)} \left[\sum_{t=0}^{T} r(s_t, a_t) + \beta \mathcal{H}(\pi_\theta)|_{s_0 \sim \rho(s_0)} \right] .$$

Preliminary

Max-entropy RL (Soft RL)

Suppose we have per-token reward $r(s_t, a_t)$, initial state distribution $\rho(s_0)$, then the KL-constrained RL objective is

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{a_t \sim \pi_\theta(\cdot|s_t)} \left[\sum_{t=0}^{T} r(s_t, a_t) + \beta \mathcal{H}(\pi_\theta) |_{s_0 \sim \rho(s_0)} \right] .$$

The optimal value function and Q-function are defined as

$$V^\star(s_t) := \underbrace{\beta \log \sum_{a_t} \exp(Q^\star(s_t, a_t)/\beta)}_{\text{a soft version of taking maximum}},\ Q^\star(s_t, a_t) := r(s_t, a_t) + V^\star(s_{t+1}) .$$

RLHF in max-entropy RL

Let the per-token reward be

$$r(s_t, a_t) = \begin{cases} \beta \log \pi_{\text{ref}}(a_t|s_t), & \text{if } s_{t+1} \text{ is not terminal} \\ r(x, y) + \beta \log \pi_{\text{ref}}(a_t|s_t), & \text{if } s_{t+1} \text{ is terminal} . \end{cases}$$

RLHF in max-entropy RL

Let the per-token reward be

$$r(s_t, a_t) = \begin{cases} \beta \log \pi_{\text{ref}}(a_t|s_t), & \text{if } s_{t+1} \text{ is not terminal} \\ r(x, y) + \beta \log \pi_{\text{ref}}(a_t|s_t), & \text{if } s_{t+1} \text{ is terminal .} \end{cases}$$

The max-entropy objective is

$$\begin{aligned} &\max_{\pi_\theta} \mathop{\mathbb{E}}_{a_t \sim \pi_\theta(\cdot|s_t)} \left[\sum_{t=0}^{T} r(s_t, a_t) + \beta \mathcal{H}(\pi_\theta) |_{s_0 \sim \rho(s_0)} \right] \\ = &\max_{\pi_\theta} \mathop{\mathbb{E}}_{x \sim \rho(x)} \mathop{\mathbb{E}}_{y \sim \pi_\theta} [r(x, y) + \beta \log \pi_{\text{ref}}(y|x) - \beta \log \pi_\theta(y|x)] \\ = &\max_{\pi_\theta} \mathop{\mathbb{E}}_{x \sim \rho(x)} \mathop{\mathbb{E}}_{y \sim \pi_\theta} [r(x, y) - \beta \, \text{KL}(\pi_\theta \| \pi_{\text{ref}})] \end{aligned}$$

RLHF in max-entropy RL

For RLHF in max-entropy RL we have the recursion

$$V^\star(x, y_{1\dots t-1}) = \beta \log \sum_s \exp(Q^\star(s|x, y_{1\dots t-1})/\beta) \ ,$$

$$Q^\star(y_t|x, y_{1\dots t-1}) = \begin{cases} \beta \log \pi_{\text{ref}}(y_t|x, y_{1\dots t-1}) + r(x, y_{1\dots t}) & y_t = \langle \text{eos} \rangle \\ \beta \log \pi_{\text{ref}}(y_t|x, y_{1\dots t-1}) + V^\star(x, y_{1\dots t}) & \text{o.w.} \end{cases} \ .$$

RLHF in max-entropy RL

For RLHF in max-entropy RL we have the recursion

$$V^\star(x, y_{1\ldots t-1}) = \beta \log \sum_s \exp(Q^\star(s|x, y_{1\ldots t-1})/\beta) \,,$$

$$Q^\star(y_t|x, y_{1\ldots t-1}) = \begin{cases} \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + r(x, y_{1\ldots t}) & y_t = \langle \text{eos} \rangle \\ \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + V^\star(x, y_{1\ldots t}) & \text{o.w.} \end{cases} \,.$$

Given an RLHF policy π, now we fix an x and $V^\star(x)$, and define

$$Q'(y_1|x) := V(x) + \beta \log \frac{\pi(y_1|x)}{1} \,,$$

$$Q'(y_2|x, y_1) := Q'(y_1|x) + \beta \log \frac{\pi(y_2|x, y_1)}{\pi_{\text{ref}}(y_1|x)} \,,$$

$$Q'(y_t|x, y_{1\ldots,t-1}) := Q'(y_{t-1}|x, y_{1,\ldots,t-2}) + \beta \log \frac{\pi(y_t|x, y_{1\ldots t-1})}{\pi_{\text{ref}}(y_{t-1}|x, y_{1\ldots t-2})} \,.$$

Q' exactly estimates $Q^\star$

(Goal) We have

$$Q(y_t|x, y_{1\ldots t-1}) = \begin{cases} \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + r(x, y_{1\ldots t}) & y_t = \langle \text{eos} \rangle \\ \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + \beta \log \sum_s \exp(Q(s|x, y_{1\ldots t})/\beta) & \text{o.w.} \end{cases}.$$

Q' exactly estimates $Q^\star$

(Goal) We have

$$Q(y_t|x, y_{1\dots t-1}) = \begin{cases} \beta \log \pi_{\text{ref}}(y_t|x, y_{1\dots t-1}) + r(x, y_{1\dots t}) & y_t = \langle\text{eos}\rangle \\ \beta \log \pi_{\text{ref}}(y_t|x, y_{1\dots t-1}) + \beta \log \sum_s \exp(Q(s|x, y_{1\dots t})/\beta) & \text{o.w.} \end{cases}.$$

(Recap) We have

$$Q'(y_t|x, y_{1,\dots,t-1}) := Q'(y_{t-1}|x, y_{1,\dots,t-2}) + \beta \log \frac{\pi(y_t|x, y_{1\dots t-1})}{\pi_{\text{ref}}(y_{t-1}|x, y_{1\dots t-2})}.$$

Q' exactly estimates $Q^\star$

(Goal) We have

$$Q(y_t|x, y_{1\dots t-1}) = \begin{cases} \beta \log \pi_{\text{ref}}(y_t|x, y_{1\dots t-1}) + r(x, y_{1\dots t}) & y_t = \langle\text{eos}\rangle \\ \beta \log \pi_{\text{ref}}(y_t|x, y_{1\dots t-1}) + \beta \log \sum_s \exp(Q(s|x, y_{1\dots t})/\beta) & \text{o.w.} \end{cases} .$$

(Recap) We have

$$Q'(y_t|x, y_{1,\dots,t-1}) := Q'(y_{t-1}|x, y_{1,\dots,t-2}) + \beta \log \frac{\pi(y_t|x, y_{1\dots t-1})}{\pi_{\text{ref}}(y_{t-1}|x, y_{1\dots t-2})} .$$

Therefore

$$Q'(y_{t-1}|x, y_{1\dots t-2}) = \beta \log \pi_{\text{ref}}(y_{t-1}|x, y_{1\dots t-2}) + \beta \log \sum_s \exp(Q'(s|x, y_{1\dots t-1})/\beta) .$$

Q' exactly estimates $Q^\star$

(Goal) We have

$$Q(y_t|x, y_{1\ldots t-1}) = \begin{cases} \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + r(x, y_{1\ldots t}) & y_t = \langle\text{eos}\rangle \\ \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + \beta \log \sum_s \exp(Q(s|x, y_{1\ldots t})/\beta) & \text{o.w.} \end{cases} .$$

(Recap) We have

$$Q'(y_t|x, y_{1\ldots,t-1}) := Q'(y_{t-1}|x, y_{1,\ldots,t-2}) + \beta \log \frac{\pi(y_t|x, y_{1\ldots t-1})}{\pi_{\text{ref}}(y_{t-1}|x, y_{1\ldots t-2})} .$$

Therefore

$$Q'(y_{t-1}|x, y_{1\ldots t-2}) = \beta \log \pi_{\text{ref}}(y_{t-1}|x, y_{1\ldots t-2}) + \beta \log \sum_s \exp(Q'(s|x, y_{1\ldots t-1})/\beta) .$$

Besides, since $\pi(y|x) \propto \pi_{\text{ref}}(y|x) \exp(r(x, y)/\beta)$, we have

$$Q'(y_t|x, y_{1\ldots t-1}) = \beta \log \pi_{\text{ref}}(y_t|x, y_{1\ldots t-1}) + r(x, y_{1\ldots t}) , \text{ when } y_t = \langle\text{eos}\rangle .$$

Your language model is secretly an advantage function

Result

Finally, we have that

$$\pi^\star(y_t|x, y_{1\dots t-1}) = \exp((Q^\star(y_t|x, y_{1\dots t-1}) - V^\star(x, y_{1\dots t-1}))/\beta) \ ,$$

where $\pi^\star$ is an optimal policy in RLHF, $Q^\star$ is the soft Q-function, and the $V^\star$ is the soft value function. The language model is implicitly an **advantage function** in max-entropy RL.

RLHF is not superior to best-of-n, intuitively

Recall the general objective of RLHF:

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x \sim \rho(x)} \mathop{\mathbb{E}}_{y \sim \pi_\theta} \left[r(x, y) - \beta \, \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}) \right] ,$$

RLHF is not superior to best-of-n, intuitively

Recall the general objective of RLHF:

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x\sim\rho(x)} \mathop{\mathbb{E}}_{y\sim\pi_\theta} \left[r(x,y) - \beta \, \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}) \right] ,$$

which is equivalent to

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x\sim\rho(x)} \mathop{\mathbb{E}}_{y\sim\pi_\theta} r(x,y) \, , \text{ w.r.t. } \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}) \leq \underset{\text{induced by duality}}{C} .$$

RLHF is not superior to best-of-n, intuitively

Recall the general objective of RLHF:

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x\sim\rho(x)} \mathop{\mathbb{E}}_{y\sim\pi_\theta} \left[r(x, y) - \beta\,\mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}})\right] ,$$

which is equivalent to

$$\max_{\pi_\theta} \mathop{\mathbb{E}}_{x\sim\rho(x)} \mathop{\mathbb{E}}_{y\sim\pi_\theta} r(x, y) , \text{ w.r.t. } \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}) \leq \underbrace{C}_{\text{induced by duality}} .$$

And this is actually what best-of-n has been doing:

Best of n

Given a policy π and x, sample $y_1, y_2, \ldots, y_n \sim \pi(\cdot|x)$, and output $y = \arg\max_{y\in\{y_i\}_{i=1}^n} r(y)$.

A truth

Goodhart's law

When a measure becomes a target, it ceases to be a good measure.

—– Charles Goodhart

A truth

Goodhart's law

When a measure becomes a target, it ceases to be a good measure.

—- Charles Goodhart

An example

For example, the goal of education is to maximize learners' learning abilities, however, when exam scores become the **target measurement**, the goal may shift to maximizing learners' exam scores. They may conflict!

A truth

Goodhart's law

When a measure becomes a target, it ceases to be a good measure.

—– Charles Goodhart

An example

For example, the goal of education is to maximize learners' learning abilities, however, when exam scores become the **target measurement**, the goal may shift to maximizing learners' exam scores. They may conflict!

So can we simply represent complex human values as reward functions?

Pessimism

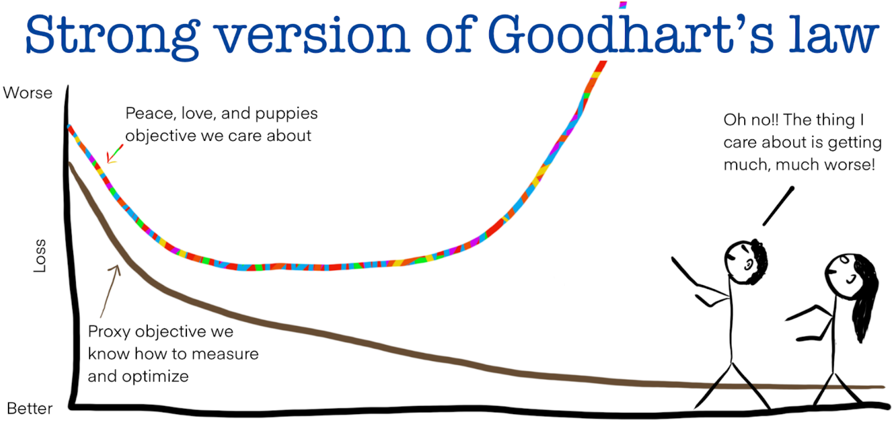
Strong Goodhart's law

Any objective, when pursued relentlessly, will eventually be misaligned.

Pessimism

Strong Goodhart's law

Any objective, when pursued relentlessly, will eventually be misaligned.



Pessimism: some examples

An example

Two sisters went to their mother's funeral. At the funeral, the younger sister saw a handsome man and fell in love with him at first sight. After returning home, the younger sister killed her older sister. Why?

Pessimism: some examples

An example

Two sisters went to their mother's funeral. At the funeral, the younger sister saw a handsome man and fell in love with him at first sight. After returning home, the younger sister killed her older sister. Why?

Another example

The Monkey's Paw by W.W.Jacobs.

Pessimism: some examples

An example

Two sisters went to their mother's funeral. At the funeral, the younger sister saw a handsome man and fell in love with him at first sight. After returning home, the younger sister killed her older sister. Why?

Another example

The Monkey's Paw by W.W.Jacobs.

AGI may grant you any wish, but how to prevent it from interpreting the wish in the most harmful way?

Even worse?

Genie:

Alien:

Grants you any wish but interprets it in the least useful / most harmful way possible

As friendly to humans as Homo Sapiens were to the Neanderthals.

Alignment still has a long way to go.

Reference

- Thoughts on AI safety. Boaz Barak's blog.
- The Crucial Role of Samplers in Online Direct Preference Optimization. arXiv 2409.19605.
- Decoding-Time Language Model Alignment with Multiple Objectives. NeurIPS 2024.
- Tuning Language Models by Proxy. COLM 2024.
- DeAL: Decoding-time Alignment for Large Language Models. ICML 2024.
- From r to $Q^\star$: Your Language Model is Secretly a Q-Function. COLM 2024.
- Weak-to-Strong Jailbreaking on Large Language Models. arXiv 2401.17256.
- A Common Pitfall of Margin-based Language Model Alignment: Gradient Entanglement. arXiv 2410.13828.