Extragradient Preference Optimization (EGPO): Beyond Last-Iterate Convergence for Nash Learning from Human Feedback

Runlong Zhou[1] Maryam Fazel[1] Simon S. Du[1]

[1]University of Washington

Preliminaries

Bandits

- $\mathcal{X}$: prompt space $\leftrightarrow$ contexts
- $\mathcal{Y}$: response space $\leftrightarrow$ actions
- Results can generalize to $|\mathcal{X}| > 1$, so omit $\mathcal{X}$ for simplicity.

Policy

- Under the *tabular softmax parametrization* common in previous works, π is parameterized by $\theta \in \mathbb{R}^{|\mathcal{Y}|}$: for any $y \in \mathcal{Y}$,

$$\pi_\theta(y) = \frac{\exp(\theta_y)}{\sum_{y' \in \mathcal{Y}} \exp(\theta_{y'})}.$$

Reinforcement learning from human feedback (RLHF)

- Given an implicit reward oracle $r : \mathcal{Y} \to [0, 1]$, Bradley-Terry (BT) model assume that human preference $\mathcal{P} : \mathcal{Y} \times \mathcal{Y} \to \Delta(\{0, 1\})$ satisfies:

$$\mathcal{P}(y_1 \succ y_2) = \sigma\left(r(y_1) - r(y_2)\right), \quad \text{where} \quad \sigma(t) = \frac{1}{1 + \exp(-t)}.$$

  Response y_1 is favored over y_2 with probability $\mathcal{P}(y_1 \succ y_2)$ by human annotators.
- Human preference dataset $\mathcal{D} = \{(y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$: in the i^{th} sample, $y_w^{(i)} \succ y_l^{(i)}$ (outcome sampled from $\mathcal{P}$).
- Learning reward r_ϕ:

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

- Learning policy regularized by on a reference policy π_{ref}:

$$\pi_\phi^\star = \arg\max_{\pi \in \Pi} \mathbb{E}_{y \sim \pi}[r_\phi(y) - \beta \mathsf{KL}(\pi || \pi_{\text{ref}})].$$

Core limitation: *transitiveness* on *population* preference.

- Even when *individual* preferences are transitive:
  Person 1: $C \succ A \succ B$, Person 2: $A \succ B \succ C$, Person 3: $B \succ C \succ A$.
  $\mathcal{P}(A \succ B) = \mathcal{P}(B \succ C) = \mathcal{P}(C \succ A) = \frac{2}{3} > \frac{1}{2}$.

Nash Learning from Human Feedback (NLHF)

RLHF as a two-player constant-sum matrix game

- Only requirement on $\mathcal{P}$: $\mathcal{P}(y \succ y') + \mathcal{P}(y' \succ y) = 1$.
- Define

$$\mathcal{P}(y \succ \pi') := \mathbb{E}_{y' \sim \pi'} \mathcal{P}(y \succ y') = \mathcal{P}\pi',$$
$$\mathcal{P}(\pi \succ \pi') := \mathbb{E}_{y \sim \pi, y' \sim \pi'} \mathcal{P}(y \succ y') = \pi^\top \mathcal{P} \pi'.$$

- Under regularization, find a policy $\pi^\star$ that is preferred over any other (adversarial) policy:

$$V_\beta(\pi_1, \pi_2) := \pi_1^\top \mathcal{P} \pi_2 - \beta \mathsf{KL}(\pi_1 || \pi_{\text{ref}}) + \beta \mathsf{KL}(\pi_2 || \pi_{\text{ref}}),$$
$$\theta_1^\star = \arg\max_{\theta_1} \min_{\theta_2} V_\beta(\pi_1, \pi_2).$$

Find the Nash equilibrium of $\mathcal{P}$!

- Due to $\mathcal{P} + \mathcal{P}^\top = \mathbb{1}$, NE satisfies

$$\theta = \theta_{\text{ref}} + \frac{\mathcal{P}\pi_\theta}{\beta}.$$

Algorithm: EGPO

- In tabular form:

$$\theta^{(t+1/2)} = (1 - \eta\beta)\theta^{(t)} + \eta\beta\left(\theta_{\text{ref}} + \frac{\mathcal{P}\pi^{(t)}}{\beta}\right),$$
$$\theta^{(t+1)} = (1 - \eta\beta)\theta^{(t)} + \eta\beta\left(\theta_{\text{ref}} + \frac{\mathcal{P}\pi^{(t+1/2)}}{\beta}\right).$$

- Neural networks:

$$\theta^{(t+1/2)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}_{\mathsf{IPO}}(\theta^{(t)}; \mathsf{Uniform}(\mathcal{Y}), \mathsf{sg}[\pi^{(t)}]),$$
$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}_{\mathsf{IPO}}(\theta^{(t)}; \mathsf{Uniform}(\mathcal{Y}), \pi^{(t+1/2)}).$$

Here we use a generalized IPO loss:

$$\mathcal{L}_{\mathsf{IPO}}(\theta; \rho, \mu) = \mathbb{E}_{(y,y') \sim \rho}\left[\left(\log \frac{\pi_\theta(y)\pi_{\text{ref}}(y')}{\pi_\theta(y')\pi_{\text{ref}}(y)} - \frac{1}{\beta}\mathbb{E}_{y'' \sim \mu}[\mathcal{P}(y \succ y'') - \mathcal{P}(y' \succ y'')]\right)^2\right].$$

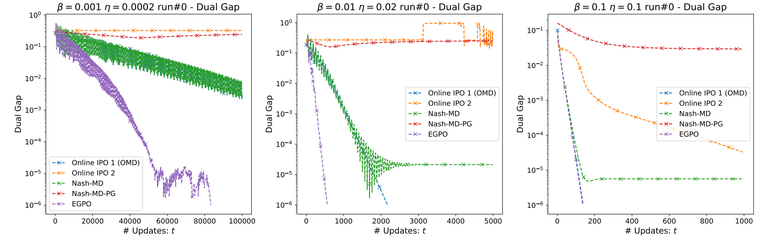
Check out our paper for equivalence using IPO (another core finding) and approximating $\mathsf{Uniform}(\mathcal{Y})$

Theoretical Results

Algorithm	Convergence to Regularized QRE	Last-iterate Convergence	Convergence to Original ε-NE
Online Mirror Descent	$\widetilde{O}(1/T)$	No	$\widetilde{O}(1/\varepsilon^2)$ iterations
Nash-MD / MTPO	$\widetilde{O}(1/T)$	**Yes**	Not provided
SPO / SPPO	Not provided	No	$\widetilde{O}(1/\varepsilon^2)$ iterations
INPO	$\widetilde{O}(1/T)$	**Yes**	Not provided
MPO	$\widetilde{O}((\frac{1}{1+\eta\beta})^T)$ **(linear)**	**Yes**	$\widetilde{O}(1/\varepsilon^2)$ iterations
ONPO	Not provided	No	$\widetilde{O}(1/\varepsilon)$ **iterations**
EGPO	$\widetilde{O}((1-\eta\beta)^T)$ **(linear)**	**Yes**	$\widetilde{O}(1/\varepsilon)$ **iterations**

- Last-iterate convergence is necessary when deploying LLMs

Simulation Results



Benchmark Results

ALG	Ep	π_{ref}	OIPO1		OIPO2		NMD		NMDPG		MPO		EGPO	
			6	8	6	9	8	10	4	8	7	8	5	8
OIPO1	6	**72.8%**			**58.6%**	**57.6%**	47.7%	46.4%	**68.4%**	**69.4%**	45.2%	47.0%	42.6%	42.8%
	8	**71.8%**			**58.9%**	**58.7%**	48.1%	47.0%	**68.2%**	**68.0%**	45.7%	47.2%	42.1%	43.6%
OIPO2	6	**66.8%**	41.4%	41.1%			39.8%	38.5%	**62.3%**	**61.3%**	41.3%	42.8%	33.8%	35.2%
	9	**66.3%**	42.4%	41.3%			38.5%	38.7%	**61.2%**	**61.3%**	40.8%	42.7%	34.2%	33.8%
NMD	8	**72.8%**	**52.3%**	**51.9%**	**60.2%**	**61.5%**			**70.0%**	**71.1%**	46.4%	48.3%	44.0%	46.7%
	10	**72.9%**	**53.6%**	**53.0%**	**61.5%**	**61.3%**			**70.6%**	**71.2%**	47.3%	49.2%	44.6%	45.8%
NMDPG	4	**55.2%**	31.6%	31.8%	37.7%	38.8%	30.0%	29.4%			31.5%	33.2%	26.2%	26.4%
	8	**55.1%**	30.6%	32.0%	38.7%	38.7%	28.9%	28.8%			31.1%	32.2%	26.2%	25.8%
MPO	7	**71.9%**	**54.8%**	**54.3%**	**58.7%**	**59.2%**	**53.6%**	**52.7%**	**68.5%**	**68.9%**			49.4%	47.9%
	8	**70.2%**	**53.0%**	**52.8%**	**57.2%**	**57.3%**	**51.7%**	**50.8%**	**66.8%**	**67.8%**			47.2%	46.9%
EGPO	5	**76.9%**	**57.4%**	**57.9%**	**66.2%**	**65.8%**	**56.0%**	**55.4%**	**73.8%**	**73.8%**	**50.6%**	**52.8%**		
	8	**77.4%**	**57.2%**	**56.4%**	**64.8%**	**66.2%**	**53.3%**	**54.2%**	**73.6%**	**74.2%**	**52.1%**	**53.1%**		

- Safe-RLHF benchmark
- Pair-wise win-rates among top-2 checkpoints from each algorithm
- NMDPG is the official implementation of Nash-MD, while NMD is our IPO implementation

vectorzh@cs.washington.edu