RLHF/DPO 小話（その２）

和地 瞭良 / *Akifumi Wachi*

April 21, 2024

RLHF [1]や DPO [2]に関して論文 [Wachi et al., 2024] を最近執筆した。やはり自分で研究すると理解がかなり深まるわけで、巷ではあまり議論されていないことが色々と分かった。なので、すでに公開済みの内容の中から話せる範囲で情報共有して、誰かの役に立てばいいな、と思っている次第である。やる気が持続すればその 4 くらいまでいく予定（多分力尽きる）。

[1] RLHF = Reinforcement Learning from Human Feedback [Ouyang et al., 2022]

[2] DPO = Direct Preference Optimization [Rafailov et al., 2024]

これまでのコラム（その１）

- `https://akifumi-wachi-4.github.io/website/jp.html`

最適解の骨格・外形はどれも同じ

その１で、RLHF であれ DPO であれ（ΨPO/IPO/KTO も）、「解いているのはすべて同じ問題だ」と述べた。復習になるが、以下の (1)

$$\max_{\pi_\theta} \quad \mathbb{E}_{x\sim\rho, y\sim\pi_\theta(\cdot|x)}\left[\, r(x,y)\,\right] - \beta \mathbb{D}_{\mathrm{KL}}\left[\, \pi_\theta(y \mid x) \,\|\, \pi_{\mathrm{ref}}(y \mid x)\,\right], \quad (1)$$

を解いているのは共通である、というわけである。これまた前回の復習になるが、(1) の最適解は解析的に書くことができ、 [3]

$$\pi_r^\star(y \mid x) = \frac{1}{Z_r(x;\pi_{\mathrm{ref}})}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right), \quad (2)$$

なる関係を得る。

[3] 骨格・外形自体を変えるという話も存在する。これについては次回解説する予定である（個人的にはこれがおもしろい）。

いま、(2) をより詳細に眺めてみよう。この式の「意味」を考えてみると、以下のように解釈できる。

- **報酬関数 r に関して言語モデルをアライメントしたときの最適解 $\pi_r^\star$ は、リファレンス方策 π_{ref} に対して $\exp\left(\frac{1}{\beta}r(x,y)\right)$ を掛けたものである。**
- **ただし、すべての $x \in \mathcal{X}$ に対し、$0 \le \pi_r^\star(y \mid x) \le 1$ かつ $\sum_y \pi_r^\star(y \mid x) = 1$ となるように分配関数 Z_r で正規化する。**

そして、(2) は、(1) を解くと決めた段階で、一切の仮定を置くことなくすべて等号で導かれた式である。したがって、**RLHF/DPO/ΨPO/IPO/KTO は、すべて (1) を解いているわけで、目指している解の骨格・外形は同じである、ということを運命づけられている**ことになる。 [4]

[4] RLHF については、(2) を用いずに (1) を強化学習で解くため、DPO/ΨPO/IPO/KTO と比較して一段と難しいことをしているともみなせる。しかし、目指している解は同じであり、強化学習アルゴリズムが完璧に動作すれば同じ解が理想的には得られる（そんなことは普通無理だが）。

どこで分岐が生じるのか？

RLHF とそれ以外については、その 1 で述べた通り (2) の関係性を用いるかどうか、という大きな分岐があった。では、(2) を用いる直接法の中で、なぜ様々なアルゴリズムが登場しているのだろうか？

正解は、

- 「報酬関数に関する仮定の違いに応じてアルゴリズムが派生している」

である。逆に言うと、DPO/ΨPO/IPO/KTO といったアルゴリズムは、(2) という最適解の外形が分かっている、という前提のもとで、**（人間が内在的に持つ）真の報酬関数 $r^\star$ に対し、いかに妥当な仮定をおいて π_{ref} の形を $\exp\left(\frac{1}{\beta}r(x,y)\right)$ を掛けて変形させるか**、というのを競い合っていることになる。

データセットと報酬構造

アルゴリズム	データセット	報酬関数に関する仮定
RLHF	Preference データ	Bradley-Terry モデル
DPO	Preference データ	Bradley-Terry モデル
ΨPO	Preference データ	Bradley-Terry モデルの一般化
IPO	Preference データ	ΨPO の特殊なケース（$\Psi(q) = q$）
KTO	Unpaired な二値データ	Kahneman-Tversky モデル

Table 1: アルゴリズムの分類

「報酬関数に関する仮定をなににするかでアルゴリズムが派生している」と述べたが、これには「どのようなデータセットが得られると仮定するか」と密接な関わりをもつ。Table 1に主要なアルゴリズムについてまとめたので参照されたい。[5] [6]

Preference data と *Bradley-Terry* モデル

RLHF と DPO は Table 1 に示す通り、データセットに関しても報酬関数に関しても全く同じ仮定を取っている。まず、データセットに関しては、

$$\mathcal{D} := \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N, \tag{3}$$

という preference data の存在を仮定する。ただし、あるプロンプト x に対し、y_w と y_l はそれぞれ、より好まれた回答（winner）とそうでない回答 (loser) である。

このとき、人間がラベル付けしたデータ $\mathcal{D}$ から、背後に潜む報酬関数 $r^\star$ を推定するわけだが、このとき Bradley-Terry モデル [Bradley and Terry, 1952] に従うのが一般的である。[7]

$$p^\star(y_w \succ y_l \mid x) = \frac{\exp\left(r^\star(x, y_w)\right)}{\exp\left(r^\star(x, y_w)\right) + \exp\left(r^\star(x, y_l)\right)} \tag{4}$$

$$= \sigma\big(r^\star(x, y_w) - r^\star(x, y_l)\big), \tag{5}$$

ただし、σ はシグモイド関数である。また、$r^\star$ は人間の潜在的な報酬関数であり、$p^\star(y_w \succ y_l \mid x)$ は、「プロンプト x に対して人間がどの程度 y_w を y_l より好むか」を示す確率である。

[5] 最近雨後の筍のように色々なアルゴリズムがポコポコ誕生しているのだが、全部紹介していると本当に話したいその 3・4 までたどり着かずに力尽きるし、そこまで重要なアルゴリズムが登場しているわけでもないと思っているので割愛。

[6] ORPO [Hong et al., 2024] は注目しておいたほうが良いかもしれない。大抵、SFT → {RLHF, DPO, KTO} のように学習して、π_{ref} をリファレンス方策にするのだが、ORPO はリファレンス方策がそもそも必要ない構造になっている。

[7] Bradley-Terry モデルは、$r^\star(x, y_w) \gg r^\star(x, y_l)$ のとき、$p^\star(y_w \succ y_l \mid x) \approx 1$ となり、$r^\star(x, y_w) \ll r^\star(x, y_l)$ のとき、$p^\star(y_w \succ y_l \mid x) \approx 0$ と滑らかに変化するようなモデルである。ボードゲームの勝ち負けなど、「比較」に基づくデータの解析に昔からよく用いられる。

RLHF

ここまでくれば、RLHF の説明はもはや終わったも同然である。まず、最初のステップ（reward modeling フェーズと呼ぶ）では、Preference data $\mathcal{D}$ が Bradley Terry モデルに従うと仮定して、

$$\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[-\log\sigma\big(r_\psi(x,y_w)-r_\psi(x,y_l)\big)\right] \tag{6}$$

という、negative loglikelihood を最小化することによって、報酬モデル $r_\psi : \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ を得る。その後、PPO や REINFORCE といったアルゴリズムを用いて

$$\max_{\pi_\theta}\quad \mathbb{E}_{x\sim\rho,y\sim\pi_\theta(\cdot|x)}\left[\,r_\psi(x,y)\,\right]-\beta\mathbb{D}_{\mathrm{KL}}[\,\pi_\theta(y\mid x)\,\|\,\pi_{\mathrm{ref}}(y\mid x)\,],$$

を解く. (1) と違い、r が r_ψ になっていることに注意されたい。[8]

[8] (1) の解釈についてはその 1 でじっくり解説しているのでそちらを参照されたい。

DPO

DPO は、RLHF とは異なり、(2) と最適解が書けることを陽に用いた手法である。いま、(2) を変形すると

$$r(x,y)=\beta\log\frac{\pi_\theta(y\mid x)}{\pi_{\mathrm{ref}}(y\mid x)}+\beta\log Z_r(x;\pi_{\mathrm{ref}}) \tag{7}$$

を得る。(2) および (7) の唯一不都合な点として、Z_r が

$$Z_r(x;\pi_{\mathrm{ref}}):=\sum_y\pi_{\mathrm{ref}}(y\mid x)\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{8}$$

というように、$\sum_y$ によって定義されている点にある。これは、あるプロンプト x に対し、π_{ref} が生成しうるあらゆる出力 y に対する和であり、計算するのは容易ではない。

しかし、美しいことに、Bradley Terry モデルの仮定の元、報酬関数を推定する際には、(6) 式のように、

$$r(x,y_w)-r(x,y_l) \tag{9}$$

という、より好まれた回答 y_w と そうでなかった回答 y_l との間の報酬の差だけ分かれば良い。いま、$Z_r(x;\pi_{\mathrm{ref}})$ は y に依存しない関数であることに注意すると、

$$\begin{aligned}
&r(x,y_w)-r(x,y_l)\\
&=\beta\log\frac{\pi_\theta(y_w\mid x)}{\pi_{\mathrm{ref}}(y_w\mid x)}+\cancel{\beta\log Z_r(x;\pi_{\mathrm{ref}})}-\beta\log\frac{\pi_\theta(y_l\mid x)}{\pi_{\mathrm{ref}}(y_l\mid x)}-\cancel{\beta\log Z_r(x;\pi_{\mathrm{ref}})}\\
&=\beta\log\frac{\pi_\theta(y_w\mid x)}{\pi_{\mathrm{ref}}(y_w\mid x)}-\beta\log\frac{\pi_\theta(y_l\mid x)}{\pi_{\mathrm{ref}}(y_l\mid x)}
\end{aligned}$$

というふうに、$Z_r(x;\pi_{\mathrm{ref}})$ の項を削除することができる。この式を (6) に代入して得るのが、DPO が結果的に最小化する損失関数であり、以下の

ように書くことができる。

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

これは、与えられたデータ $\mathcal{D}$ のもとで、尤度を最大化しているだけである。「報酬モデルの学習」 → 「強化学習」を行う必要のある RLHF によりはるかにシンプルであるのが分かるだろう。[9]

[9] 実用上も、計算時間・安定性などの面で DPO が優れている。一方で、RLHF の方が結果的に得られる言語モデルの性能は高い、という報告 [Ahmadian et al., 2024] もあり、RLHF と DPO の優劣は依然として議論の余地がある。ここに関しては私も本当に分からない。

RLHF と *DPO* の等価性

いま一度、RLHF と DPO の関係性について議論しよう。まず、データセットと報酬関数に関する仮定は、完全に同じである。データセットは Preference data であり、報酬関数も Bradley-Terry Model に従うことを仮定している。解いている問題についても、(1) という完全に同じ問題を解いている。では、(2) や (7) を用いているという点についてはどうであろうか？これらは、(1) の厳密な最適解であり、導出の仮定で一切の仮定や近似なしに、すべて等式で導かれたものである [10]。したがって、RLHF と DPO は、手順こそ違うものの、同じ問題設定のもと同じ最適解を目指した手法であると言える。これが、**「RLHF と DPO は等価である」**と言われる所以である。もちろん、両方の手法において最適解は基本的には得られないので、実際に得られる解は当然異なることに注意が必要である。

[10] 導出についてはその 1 参照。

Ψ*PO*

ΨPO [Azar et al., 2023] は、DPO を一般化したものとして提案されたアルゴリズムである。ΨPO も、RLHF や DPO と同じ Preference data

$$\mathcal{D} := \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N,$$

を用いて学習を行う。異なる点としては、$\Psi : [0,1] \to \mathbb{R}_+$ なる非減少関数を導入して、真の報酬関数が

$$r^\star(x,y) = \mathbb{E}_{y'\sim\pi_{\mathrm{ref}}}\Psi(p^\star(y \succ y' \mid x)) \tag{10}$$

という式に従うと仮定していることである。くどいようだが、RLHF や DPO と比較し、ΨPO は、データセットの形式は同じ、最適化すべき式はともに (1) で同じ、報酬関数に関する仮定だけが異なる、という点に注意されたい。

実はこの形は、ΨPO の (10) は、RLHF や DPO で用いられた Bradley Terry モデル の一般化になっている。なかなか面白いので式を見ていこう。いま、関数 Ψ を

$$\Psi(q) = \log\left(\frac{q}{1-q}\right) \tag{11}$$

と定義する。いま、Bradley-Terry モデルに従う

$$p^\star(y \succ y' \mid x) = \sigma(R(x,y) - R(x,y')) \tag{12}$$

なる報酬関数 $R : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ が存在するとする。このとき、(10) は、

$$\begin{aligned}
\mathbb{E}_{y' \sim \pi_{\text{ref}}} \Psi(p^\star(y \succ y' \mid x)) &= \mathbb{E}_{y' \sim \pi_{\text{ref}}} \Psi(p^\star(y \succ y' \mid x)) \\
&= \mathbb{E}_{y' \sim \pi_{\text{ref}}} \Psi\left(\frac{\exp(R(x,y))}{\exp(R(x,y)) + \exp(R(x,y'))}\right) \\
&= \mathbb{E}_{y' \sim \pi_{\text{ref}}} \log\left(\frac{\exp(R(x,y))}{\exp(R(x,y'))}\right) \\
&= \mathbb{E}_{y' \sim \pi_{\text{ref}}}[R(x,y) - R(x,y')] \\
&= R(x,y) - \mathbb{E}_{y' \sim \pi_{\text{ref}}}[R(x,y')]
\end{aligned}$$

となる。右辺第二項は定数なので、この報酬関数を (1) に代入したとしても、最適方策には影響を与えない。すると、実質的には報酬関数 R で方策を最適化していることになり、これはまさしく DPO が行っていることと等価である。

また、Azar et al. [2023] では、ΨPO の別の特殊ケースとして IPO を提案している。これは $\Psi(q) = q$ という恒等関数（Identity function）を用いるものである。

KTO

今まで解説した RLHF/DPO/ΨPO/IPO はすべて、Preference data を用いたものであった。ここでいよいよデータセットの形式にメスが入ることになる。

Preference データというのは、同一のプロンプト x に対し、2 つ（以上）の出力 y_1 と y_2 を用意して人間がその優劣をラベル付けするものであった。これはデータ収集のコストが多く、可能であれば

- プロンプトと単一の出力 y のペア (x, y) が良いかどうかのラベル

を含んだデータセットのほうが容易に集めることができる。

KTO は、プロスペクト理論 Kahneman and Tversky [1979] [11] に着想を得た手法である。例えば以下のような状況を考えよう。

- 当たりとハズレが 50%ずつのくじがあるとする。
- 当たりが出た場合は 200 万円もらえる
- はずれが出た場合は 100 万円を支払わなければならない。

論理的に考えれば、このくじの期待値は 50（=200 × 0.5 - 100 × 0.5）万円なので、くじを引くべきである。しかし、「参加しない」という選択をする人が以外にも多い、というのがプロスペクト理論の概略である。[12] これは、人が利得よりも損失を大きく感じる傾向をもち、損失を 2.25

[11] Kahneman と Tversky がノーベル賞を受賞した経済学の理論。なんかこういうの良いよね

[12] プロスペクト理論自体に関しては素人なので間違えているかも

倍強く感じると推定されている。[13]

KTO においては、ある基準点となる報酬関数の値からの差分を、代理的な報酬とみなす。報酬関数の基準値は大抵の場合 π_{ref} を用いて計算され、

$$\nu := \beta\, \mathbb{D}_{\text{KL}}[\, \pi_\theta(y \mid x) \,\|\, \pi_{\text{ref}}(y \mid x)\,]$$

と定義される。[14]

プロンプト x と単一の出力 y によって構成されたデータセット $\widetilde{\mathcal{D}} := \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ に対し、KTO の損失関数は以下のように計算される。

$$\mathcal{L}_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}, \beta) = \mathbb{E}_{x,y\sim\widetilde{\mathcal{D}}}\,[\, v_{\text{KTO}}(x, y, \beta)\,], \tag{13}$$

いま、v_{KTO} は価値関数と呼ばれ、ν とひかくしたときの潜在的な報酬関数 $r_{\text{KTO}}(x,y) := \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ の値を用いて以下のように定義される。

$$v_{\text{KTO}}(x, y, \beta) := \begin{cases} w_+(1 - \sigma(r_{\text{KTO}}(x,y) - \nu)) & \text{if} \quad y \sim y_+ \mid x \\ w_-(1 - \sigma(\nu - r_{\text{KTO}}(x,y))) & \text{if} \quad y \sim y_- \mid x. \end{cases}$$

なお、y_+ と y_- はそれぞれ望ましい回答とそうでない回答であり、w_+ と w_- はそれぞれに対する重みである。[15]

[13] ラベルがそこまで緊迫感を持って付けられているのかは結構謎。ユーザーフィードバックを学習に使いやすくなる、みたいな文脈だとそんなに「損失」感じるか？とか思ったりはする（勘違いしてたら誰か教えて）

[14] その 1 でも述べたが、$\mathbb{D}_{\text{KL}}$ を Reverse KL divergence の期待値（$\mathbb{E}_{x\sim\rho}$）として定義していることに注意

[15] KTO は Preference data を前提としていないので、それぞれの回答が「絶対的に」望ましいかどうか、であることに注意。

今後書くこと（予定）

- 言語モデルのアライメントは確率分布を尖らせる？
- (1) で Reverse KL divergence を用いるのは善か悪か
- (1) と強化学習の関係
- なぜ SACPO は制約付きの方策最適化問題を stepwise に解くことが許されるのか？

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39 (3/4):324–345, 1952.

Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Akifumi Wachi, Thien Q Tran, Rei Sato, Takumi Tanabe, and Yohei Akimoto. Stepwise alignment for constrained language model policy optimization. *arXiv preprint arXiv:2404.11049*, 2024.