

RLHF/DPO から始める始める強化学習入門

Akifumi Wachi / 和地 瞭良

2024年6月22日

1 なぜこの資料を作ったか？

ChatGPT の登場とともに、RLHF¹⁾ や DPO²⁾ などの「言語モデルのアライメント」に注目が集まっている。私の書いた

- DPO 解説スライド

– https://speakerdeck.com/akifumi_wachi/neurips-23-can-jia-bao-gao-plus-dpo-jie-shuo

- コラム 『RLHF/DPO 小話』

– <https://akifumi-wachi-4.github.io/website/jp.html>

も想像以上の反響をいただき、(自称) 強化学習研究者としてはありがたい限りである。私の周りでも「RLHF/DPO の理解が深まった！」というご意見をいただくことが何回かあり、私の資料が役に立ったのであれば非常に嬉しい。

一方で、これまた何人かの方から

「強化学習を理解した！」

という風に言われた。これについてはちょっと複雑な気持ちになるわけで、RLHF や DPO を理解したくらいでは「強化学習」を理解したわけではないんだよね、と思わずにはいられない。スタバでキャラメルマキアート飲んでいる人がコーヒーを理解しているわけではないよね、という気持ちに近い？キャラメルマキアート美味しいけどさ。

かといって、新規参入者に厳しいコミュニティが衰退の一途をたどるのはゲームやスポーツなど色々な業界を見ても明らかである。強化学習というとマルコフ決定過程 (Markov decision Process, MDP) とか小難しい概念がいきなり登場して挫折しやすいわけである。でも、RLHF や DPO がここまで話題ということは、言語モデルのアライメントという特定のタスクにおける強化学習の使われ方を理解している人が一定数いるというわけで、強化学習に入門するための土台がすでにある、

1) RLHF = Reinforcement Learning from Human Feedback。なお、本資料で RLHF という名前をつかうときは、[2] や [8] において提案されている手法のことを指す。本来は、文字通り人間のフィードバックで強化学習する一般的な用語だったはずなのだが、特殊ケースが一般名詞を乗っ取ってしまった。。。

2) DPO = Direct Preference Optimization

ということになる。したがって、せっかく強化学習に興味を持ってくださっている方が多いこのタイミングで入門的な資料を作ろう、と思った次第である。

2 本資料の対象者

- 強化学習に入門したい人
- RLHF/DPO についてはそれなりに知っている人
- 一般の強化学習 と RLHF/DPO の関係性についてはあまり分からない人

本資料においては、通常の強化学習の講義や書籍とは全く異なった形で議論を進めたいと思う。具体的には、言語モデルのアライメントにおける RLHF や DPO はある程度理解している、という前提で強化学習の概念を解説するので悪しからず。

3 RLHF や DPO が解いていた問題とは

RLHF [2] や DPO [4] で解いていた問題は以下の通りである。³⁾

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

ただし、 $x \in \mathcal{X}$ はプロンプト、 $y \in \mathcal{Y}$ は言語モデル (Language Model, LM) の出力である。入力 x は何らかの確率分布 ρ からサンプルされるとする。このとき、言語モデルは入力 x に対し出力 y を返す方策とみなすことができる。いま、 π_{ref} と π_{θ} という2つの方策が (1) に存在するが、 π_{θ} は θ によってパラメタライズされた学習中の方策、 π_{ref} は元となるリファレンス方策を表す。 $r : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ は、入力 x と出力 y のペアに対して、その良し悪しを実数値で返す関数で報酬関数と呼ばれる。最後に、 \mathbb{D}_{KL} は Reverse KL divergence であり、(1) を簡潔に書くため、すでに $\mathbb{E}_{x \sim \rho}[\cdot]$ の期待値が取られていることに留意されたい。

いま、新たな報酬関数 $R : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ を

$$R(x, y) := r(x, y) - \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} \quad (2)$$

のように定義すると、(1) は

$$\max_{\pi_{\theta}} \mathbb{E}_{x \sim \rho, y \sim \pi_{\theta}(\cdot|x)} [R(x, y)] \quad (3)$$

というシンプルな形で書くことができる。

3.1 (3) の解釈

(3) を詳細に解釈してみよう。

3) これは KTO や IPO でも同様である。詳細が気になる方は、コラム『RLHF/DPO 小話』を参照のこと。

- まず、プロンプト x が ρ からサンプルされる。実際には人間がプロンプトを作成するわけだが、言語モデルの目線から見ると、プロンプト x はどこからともなく降ってくるものである。
- 言語モデルが方策 π_θ にしたがって出力 y を生成する。
- 出力 y の生成は報酬関数 R を最大化することが至上命題である。
- 報酬関数は、プロンプト x と出力 y のペア (x, y) に対して定義されるので、
 - 人間がプロンプト作る → 言語モデルが回答を返す
 という一問一答に対し、報酬が定義されていることになる。

この問題の重要な特徴だけもう一度言うと、以下の通りである。

- プロンプトはどこからともなく降ってくるもの
- プロンプトに対して回答を一回返して終わり

3.2 (1) や (3) は文脈付きバンディット問題

実は上の二個の特徴は、「一般の」強化学習が扱う問題と比較するとかなり簡略化されたものである。RLHF や DPO が解いている問題 (1) や (3) は、厳密には「文脈付きバンディット」という問題としてみなしたほうが適切である。⁴⁾

文脈付きバンディット [1] は、推薦システム等でよく使われるのでご存知の方も多いかもしれない。文脈付きバンディット問題において、アルゴリズムは、文脈 (context) $x \sim \rho$ を観測し、文脈 x に対して、報酬 $R(x, y)$ が最大となるようなアーム y を選ぶことを目指す。「文脈」は、文献によっては、特徴量 (feature) と呼ばれることも多い。ショッピングサイトの例を挙げるならば、ユーザーの属性 (年齢・性別など) が文脈 x である。アームというのは、アルゴリズム目線でみたときにユーザーの推薦するアイテムのことである。

このとき重要なのが、

- アルゴリズムは、どのユーザーに対し推薦するかを自分では選べないので、文脈 x はどこからともなく降ってくるものであること
- あるユーザーに対し、アイテムを一回推薦する状況しか考えていないこと

の二点であり、RLHF や DPO が解いていた問題の特徴と同じである。ただし、RLHF や DPO においては、

- 文脈 → プロンプト
- アーム → 回答

というように推薦システムのときは名称が変更されていることに留意されたい。すなわち、RLHF や DPO の解いている問題は、文脈 x がプロンプト、アーム y が言語モデルの出力・回答にそれぞれ置き換えられた文脈付きバンディット問題とみなすことができる。以上をまとめると、

4) 後述するが、強化学習という名称を使っても別に間違いではない。

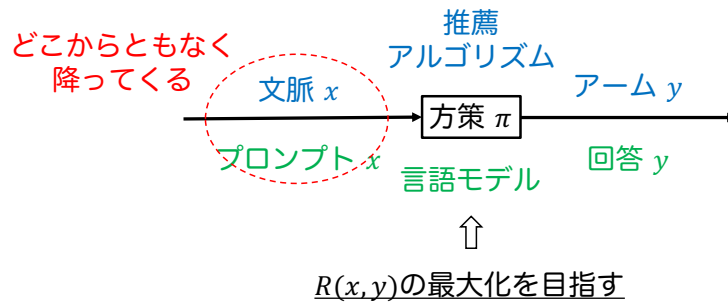


図1 文脈付きバンディットのイメージ図

図1 のようになる。

4 文脈付きバンディットから強化学習へ

強化学習と文脈付きバンディットの関係性を一言で言えば、

強化学習 \supset 文脈付きバンディット

である。まず、強化学習と文脈付きバンディットの最大の違いはなにか？文脈付きバンディットの使用語を用いるならば、文脈 x が選択したアーム y によって変化する、ということである。文脈付きバンディットにおいてはどこからともなく降ってくるものであった文脈 x が、強化学習においては自らの行動選択に応じて（ある程度）制御可能である、ということの意味する。なお、強化学習においては、「文脈」や「アーム」という用語は用いず、それぞれ「状態 (state)」・「行動 (action)」という用語を用いる。そのうえで、状態遷移確率 $P(x' | x, a)$ を定義して、ある状態 $x \in \mathcal{X}$ において行動 $y \in \mathcal{Y}$ を実行したときに、次の状態 $x' \in \mathcal{X}$ に遷移する確率を表現する。

	文脈付きバンディット	強化学習
用語	文脈 (context) ・ 特徴量 (feature) アーム (arm)	状態 (state) 行動 (action)
自らの意思決定によって変化するもの	報酬のみ	報酬 と 次の状態

表1 文脈付きバンディットと強化学習における用語の対応関係

以上をまとめると対応関係は上記の通りである。強化学習においては、自らの行動によって次の状態が遷移する、という問題設定を扱うため、文脈付きバンディットと比較してより一般的かつ複雑な問題を解いていることになる。

4.1 強化学習のメリット：損して得取れ

強化学習は、自らの行動選択が状態遷移にどのような影響を及ぼすかを考慮しながら方策を最適化する。⁵⁾我々の日常生活を考えると、強化学習的な思考をしているのではないだろうか？例えば、甘いお菓子とか毎日食べないですよ？⁶⁾なぜでしょうか？健康に悪いから・太るからですよ？でも短期的にみれば、甘いものを食べるのは幸せなわけで、そんなすぐ体重が増加したり糖尿病になったりするわけでもないの、強化学習的な用語を使うと「(即時)報酬が大きい」わけである。つまり、自分の今の意思決定が、長期的に見てどのような影響を及ぼすかを考えているのが強化学習の面白いところであり難しいポイント、ということになる。

囲碁や将棋などのAI(例: AlphaGo [5])で強化学習が使われるのもまさしくこの特徴によるものです。短期的に見れば損とも思える手でも最終的な「勝ち」につながるのであれば指さないとはいけないのが、囲碁や将棋の面白いところであろう。

4.2 期待累積報酬

強化学習の最大の特徴が、「自らの行動選択が状態遷移にどのような影響を及ぼすかを考慮しながら方策を最適化すること」であるとして、目的関数はどのように定義すべきだろうか？強化学習の問題設定で(3)を最大化したとしたらどうなるだろうか？

- 状態遷移を色々考えた挙げ句、次の時刻の報酬関数だけを最大化する

ということになるのが分かるだろうか？さながら、「甘いもの食べたら太るな」とか「来月健康診断あるしな」とか脳内で色々シミュレーションしたあげく、「よし！食べたいからケーキ食べよう」ってというようなエージェントが出来上がるわけである。

したがって、強化学習において用いる目的関数は(3)のままではだめで、将来のことも視野に入れた目的関数を定義する必要がある。このとき用いられるのが、期待累積報酬という考え方で

$$\mathbb{E} \left[\sum_{t=0}^H \gamma^t r(s_t, a_t) \mid \pi, P \right] \quad (4)$$

のように定義される。 $\gamma \in (0, 1]$ は割引率と呼ばれ、 $\{0.9, 0.95, 0.99\}$ 等の値を取ることが多い。なお、(4)は、有限区間 H における割引された期待累積報酬を考えているが、無限区間($H \rightarrow \infty$)や割引されない(すなわち $\gamma = 1$)期待累積報酬を考える場合もある。

(4)において、期待値の中を書き下してみると

$$r(s_0, a_0) + \gamma \cdot r(s_1, a_1) + \gamma^2 \cdot r(s_2, a_2) + \dots + \gamma^H \cdot r(s_H, a_H) \quad (5)$$

例えば、 $\gamma = 0.9$ かつ $H = 100$ のとき、 $\gamma^H \approx 2.7 \times 10^{-5}$ となり、実質 $\gamma^H \cdot r(s_H, a_H)$ は無視されることになる。つまり、遠い未来に得られるかもしれない報酬についてはそこまでアテにせず今後

5) 状態遷移確率を陽に推定しながら方策を最適化するモデルベース強化学習とそうでないモデルフリー強化学習があるが、いずれも自らの行動で状態遷移することを考慮しているのは共通である。

6) 一部の人に刺さったら申し訳ない。一番わかり易いかなと思ひ・・・

の行動を決めよう、という思いが込められていることになる。

強化学習の設定においては、状態遷移が確率的であり、多くの場合方策も確率的である。したがって、状態 s も確率変数になるので、(4) で $\mathbb{E}[\cdot|\pi, P]$ のように期待値がとられていることになる。

5 言語モデルのアライメントにおいて、文脈付きバンディットではなく（一般の）強化学習は必要なのか？

語弊を恐れずに野球で例えると、

- 文脈付きバンディット：ストレートだけを毎回全力で投げる
- 強化学習：緩急織り交ぜながら打者を打ち取る

のようなイメージを持っていただきたい。ロボットの制御等であれば、最終的な目標を達成するための事前動作のようなものが必要になるので、「損して得取る」ための（一般の）強化学習が必要になるだろう。

では、言語モデルのアライメントにおいて、一般の強化学習は必要なのだろうか？個人的な感想を言えば、「言語モデルにどのような挙動を求めているかによる」が正解だと思う。まず、一般の強化学習的なアプローチが必要な状況について一つ述べよう。

5.1 自然な会話をさせたい場合

たとえば、ChatGPT との対話例を以下に示す。ChatGPT に慣れた我々からすると、「いい回答してるなあ、さすが ChatGPT !!」と思うかもしれないが、これ友人や恋人との会話と考えたらかなり不自然というか、喋りすぎというか、本気出しすぎなのである。

これが、コーディングに関する話題とかであれば、とてもありがたい回答をしゅつりよくしてくれるチャットボットであろう。しかし、おしゃべりすること自体が目的であったり、文章としてではなく会話として自然なチャットボットを作りたい場合には、「160km/h のストレートを全力で投げ続ける」文脈付きバンディットの的アプローチでは限界がある、ということになる。

プロンプト

おすすめの旅行先は？

ChatGPT の回答

おすすめの旅行先は、あなたの好みや旅行の目的によって異なりますが、以下にいくつかの人気のある旅行先を紹介します。

自然を楽しみたい方へ

1. ****ニュージーランド**** - 壮大な自然景観、トレッキングコース、アウトドアアクティビティが豊富。 - ミルフォードサウンドやトンガリロ国立公園など。
2. ****アイスランド**** - 火山、氷河、温泉などのユニークな自然景観。 - ブルーラグーンやゴールデンサークルなどの観光スポット。

文化や歴史を楽しみたい方へ

1. ****イタリア**** - ローマの古代遺跡、フィレンツェのルネサンス美術、ベネチアの運河など。 - 美食やワインも魅力。
2. ****日本**** - 京都の寺社や庭園、東京の現代文化、広島歴史的遺産など。 - 季節ごとの美しい風景も楽しめます（桜や紅葉）。

リラックスやリゾートを楽しみたい方へ

1. ****モルディブ**** - 美しいビーチとクリスタルクリアな海、豪華なリゾート。 - シュノーケリングやダイビングも人気。
2. ****ハワイ**** - オアフ島やマウイ島など、さまざまなアクティビティと美しい自然。 - ビーチや火山、公園など。

都市観光を楽しみたい方へ

1. ****ニューヨーク**** - 多様な文化、エンターテインメント、ショッピングが充実。 - タイムズスクエア、セントラルパーク、ブロードウェイなど。
2. ****パリ**** - エッフェル塔、ルーヴル美術館、セーヌ川沿いの散策など。 - 美食やファッションも楽しめます。

それぞれの旅行先には、それぞれの魅力がありますので、興味や目的に合わせて選んでみてください。また、現地の気候や季節も考慮するとより充実した旅行になるでしょう。

5.2 行動をトークンとみなしてあげると？

DPO 論文においては、文脈付きバンディットの設定を考えていて、

- アーム = 言語モデルの回答

と定義していた。しかし、

- 行動 = 次トークン

と定義してあげるとする。一つの回答はたくさんのトークンの集まりなわけで、最終的な回答の質は複数の時刻ステップにわたる行動 (= トークン) の系列によって決まるからである。「行動 = 次トークン」と定義してあげると、「意味はさほどもたないが、ないと文章として成立しない助詞を出力する」などの「損して得取る」行動が必要となる。まさしく強化学習的なアプローチが必要になるわけである。

5.3 次回予告

このように、文脈付きバンディットではなく一般の強化学習を考える、言い換えるならば「行動 = トークン」とみなして言語モデルをアライメントする、という研究がここ数ヶ月登場している。今回は以下のような論文を参考にしながら、トークンレベルの言語モデルのアライメントについて解説する。

- Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to Q^* : Your language model is secretly a Q -function. arXiv preprint arXiv:2404.12358, 2024 [3]
- Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. arXiv preprint arXiv:2404.11999, 2024. [6]
- Han Zhong, Guhao Feng, Wei Xiong, Li Zhao, Di He, Jiang Bian, and Liwei Wang. DPO meets PPO: Reinforced token optimization for RLHF. arXiv preprint arXiv:2404.18922, 2024. [7]

参考文献

- [1] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.
- [2] 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 (NeurIPS)*, 2022.
- [3] Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^* : Your language model is secretly a q -function. *arXiv preprint arXiv:2404.12358*, 2024.
- [4] 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*, 2024.
- [5] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.
- [6] Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level

direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.

[7] Han Zhong, Guhao Feng, Wei Xiong, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*, 2024.

[8] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.