南京大学 人工智能学院
School of Artificial Intelligence, Nanjing University

LAMDA
Learning And Mining from DatA

Provable Efficiency in Online RL:

Function Approximation and RLHF

Peng Zhao

School of Artificial Intelligence

Nanjing University

April 29, 2025 @ NUS

NUS
National University of Singapore

Outline

- Background
- RL with Function Approximation
- RL with Human Feedback
- Conclusion

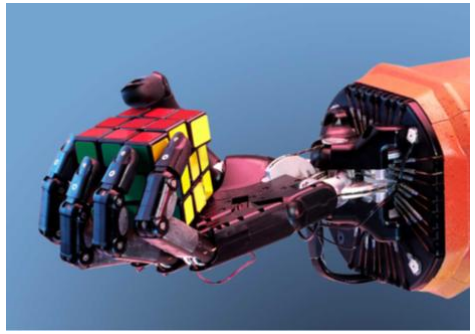
Outline

- Background
- RL with Function Approximation
- RL with Human Feedback
- Conclusion

Reinforcement Learning

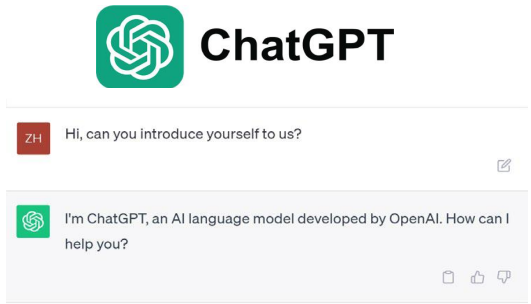
- RL has achieved great success in many applications

Games

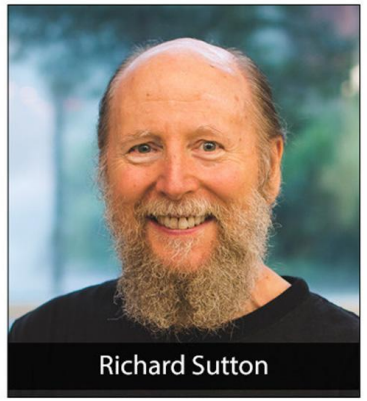


Control

Automatic Driving



Large Language Models

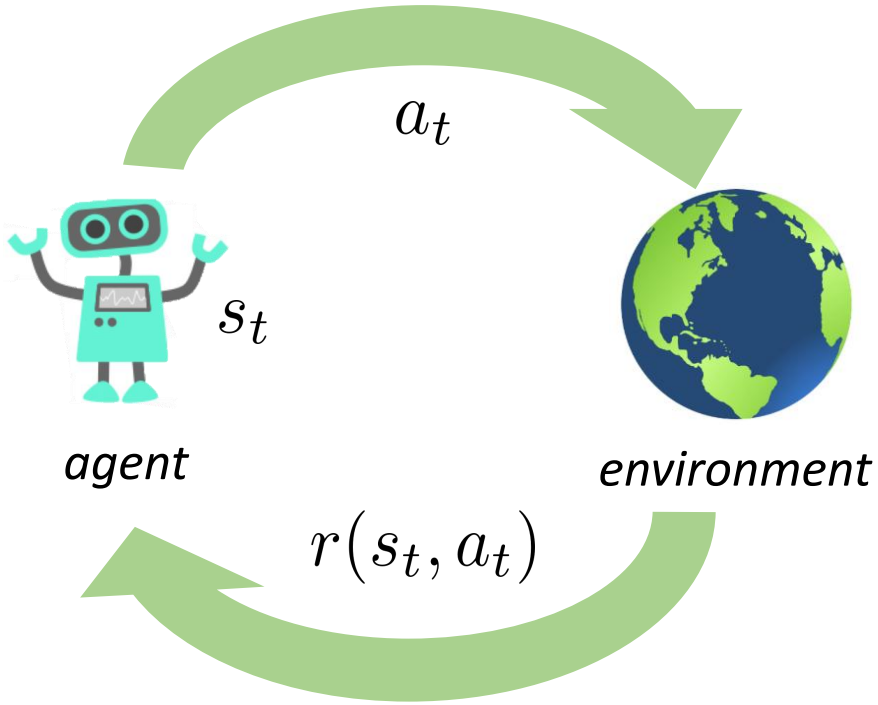


2024 ACM A.M. Turing Award

ACM A.M. Turing Award honors two researchers who led the development of cornerstone AI technology. Andrew Barto and Richard Sutton are recognized as pioneers of reinforcement learning.

Reinforcement Learning

- RL offers a principled framework for sequential decision making in ***unknown*** and ***interactive*** environments.



Supervised learning

- labeled data passively collected in advance
- minimize the cumulative loss (e.g., ERM)
- learning from examples

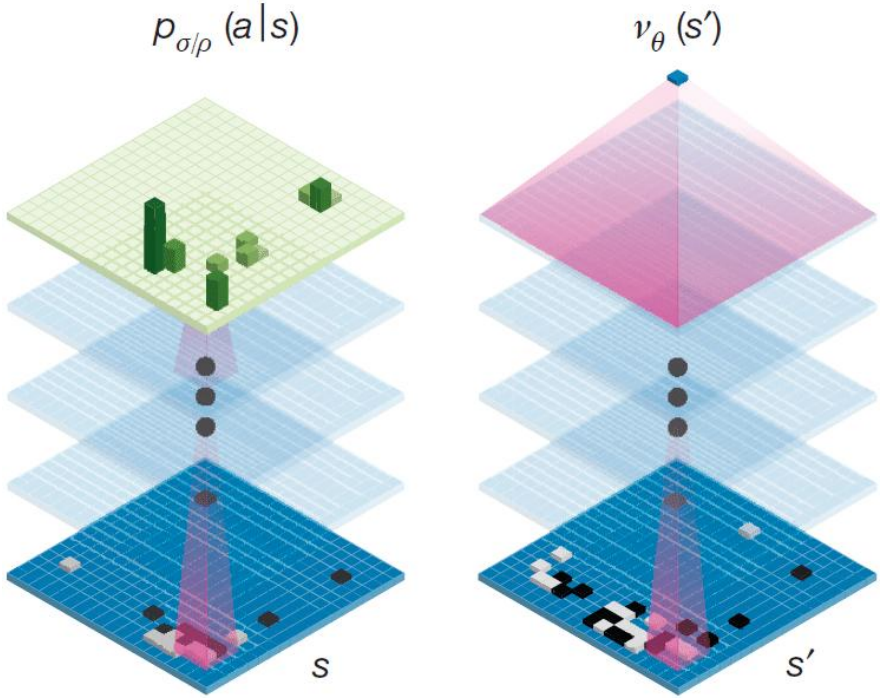
Reinforcement Learning

- agent actively interacts with the environment
- learning from feedback (rewards) to improve future behavior (doing right action).
- learning by trial-and-error

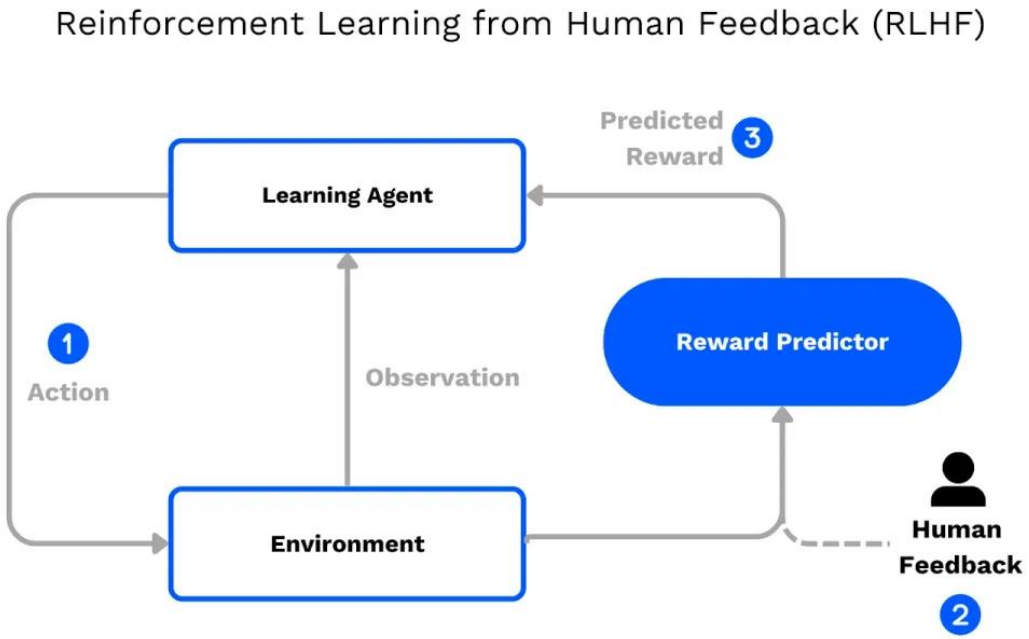
Reinforcement Learning

- (Two of) key techniques in this wave of RL success:

(i) RL with function approximation



(ii) RL from human feedback



Challenges: Efficiency

RL with function approximation		RL from human feedback	
		ChatGPT	deepseek
29 million of game *40 days of training*	*200 years of play* *44 days of training*	*13T data* *$63M cost of training*	*14.8T data* *$6M cost of training*

Goal: ***statistically*** and ***computationally*** efficient algorithm with provable guarantee.

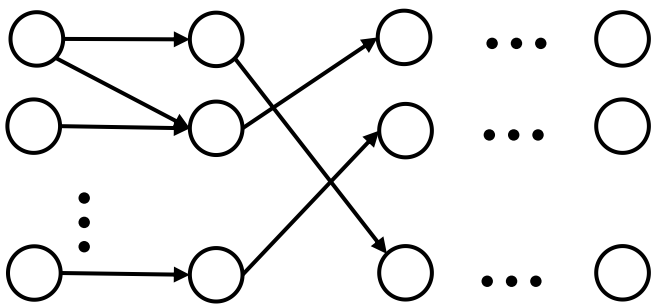
- Li, **Z**, Zhou. Provably Efficient Reinforcement Learning with Multinomial Logit Function Approximation. NeurIPS 2024.
- Li, Qian, **Z**, Zhou. Provably Efficient RLHF Pipeline: A Unified View from Contextual Bandits. Arxiv, 2502.07193.

Outline

- Background
- RL with Function Approximation
- RL with Human Feedback
- Conclusion

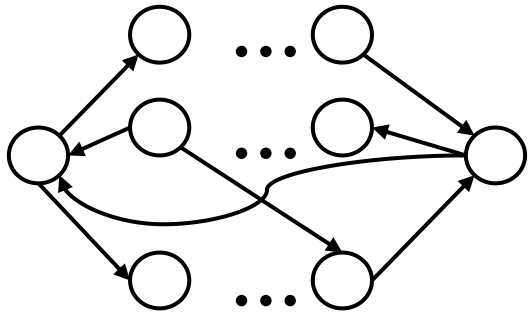
RL and MDPs

Episodic MDPs

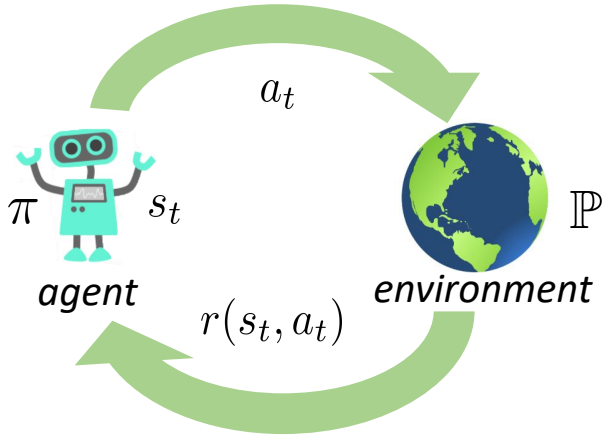


fixed steps

Infinite-horizon MDPs

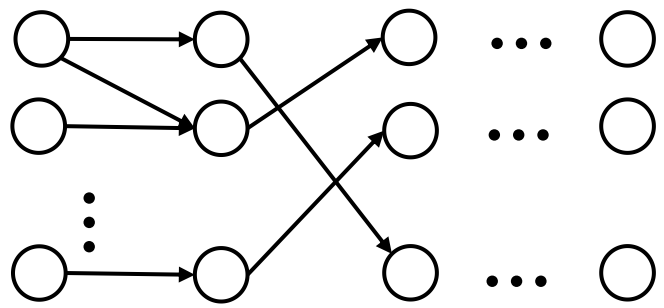


infinite steps



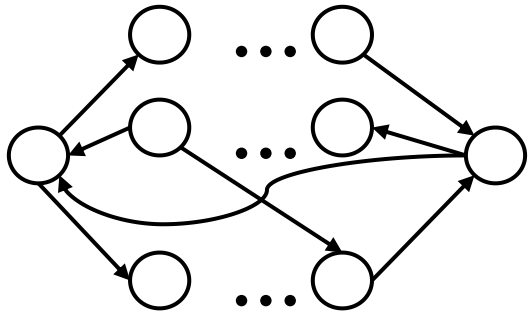
RL and MDPs

Episodic MDPs

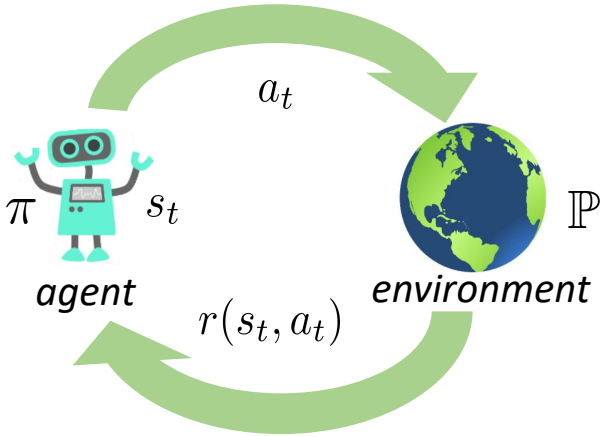


fixed steps

Infinite-horizon MDPs

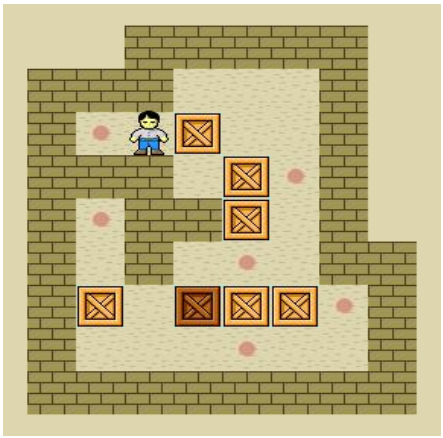


infinite steps



Episodic MDP $(\mathcal{S}, \mathcal{A}, \{\mathbb{P}_h\}_{h=1}^H, \{r_h\}_{h=1}^H)$

- $\mathcal{S}$: state space, here we consider finite $|\mathcal{S}|$.
- $\mathcal{A}$: action space, here we consider finite $|\mathcal{A}|$.
- $\mathbb{P}_h : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, transition probability.
- $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$, reward function.

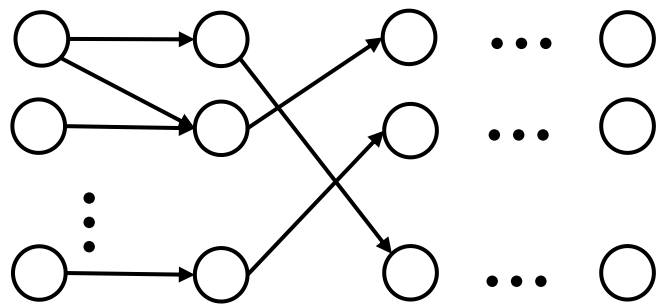


Sokoban

Pac-man

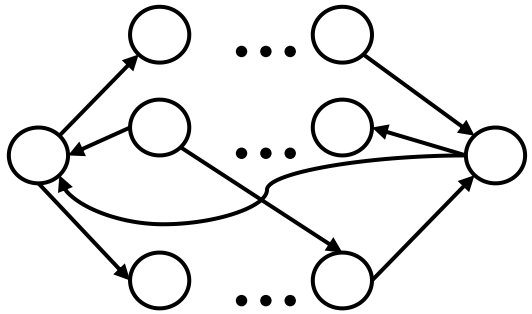
RL and MDPs

Episodic MDPs

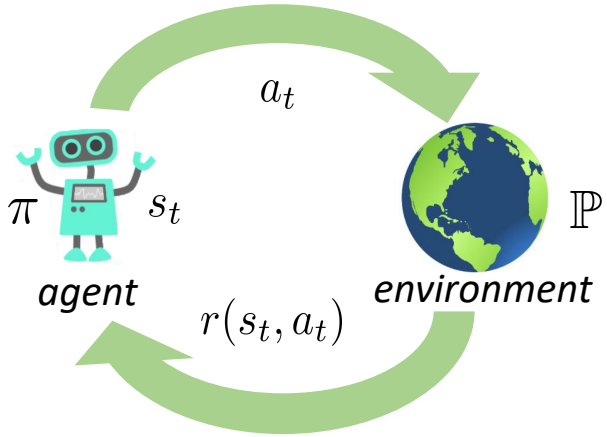


fixed steps

Infinite-horizon MDPs



infinite steps



Episodic MDP $(\mathcal{S}, \mathcal{A}, \{\mathbb{P}_h\}_{h=1}^H, \{r_h\}_{h=1}^H)$

- $\mathcal{S}$: state space, here we consider finite $|\mathcal{S}|$.
- $\mathcal{A}$: action space, here we consider finite $|\mathcal{A}|$.
- $\mathbb{P}_h : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, transition probability.
- $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$, reward function.

- Deterministic policy: $a = \pi(s)$ for $\pi : \mathcal{S} \to \mathcal{A}$.
- Stochastic policy: $a \sim \pi(s)$ for $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.

* Planning: given a fully specified MDP.

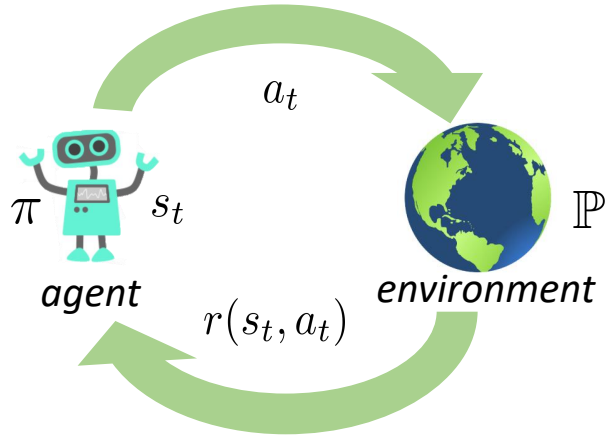
* Learning: unknown transition or reward.

Online MDPs

Online Episodic MDPs

For each episode $k = 1, \ldots, K$:

- For each stage $h = 1, \ldots H$:
  - Learner observes state $s_{k,h}$, execucts action $a_{k,h} \sim \pi_{k,h}(\cdot|s_{k,h})$, obtains reward $r_h(s_{k,h}, a_{k,h})$.
  - Learner transits to next state $s_{k,h+1} \sim \mathbb{P}_h(\cdot \mid s_{k,h}, a_{k,h})$.



$$V_h^\pi(s) = \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \mid s_h = s \right]$$

$$Q_h^\pi(s,a) = r_h(s,a) + \sum_{s'} \mathbb{P}_h(s' \mid s,a) V_{h+1}^\pi(s')$$

$$\text{Regret}_K = \max_{\pi \in \Pi} \sum_{k=1}^{K} V_{k,1}^{\pi}(s_{k,1}) - \sum_{k=1}^{K} V_{k,1}^{\pi_k}(s_{k,1})$$

to learn as well as the best policy in hindsight

Focus on ***known reward*** and ***unknown transition*** as learning reward is no harder than transition.

Challenge of Large-scale MDPs

- How to design RL algorithm to handle ***large-scale*** MDPs ?

Figure 1

We discover through experience that this state is bad

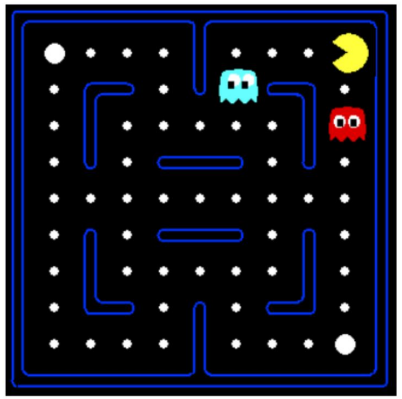


Figure 2

In tabular methods, we know nothing about this state.

Figure 3

*We know **nothing** about this state either!*

- **Tabular MDPs**: usually maintain a table to store values for all states (or state-action pairs), which scales with state number S and action number A.

Challenge of Large-scale MDPs

- How to design RL algorithm to handle ***large-scale*** MDPs ?

Figure 1

We discover through experience that this state is bad

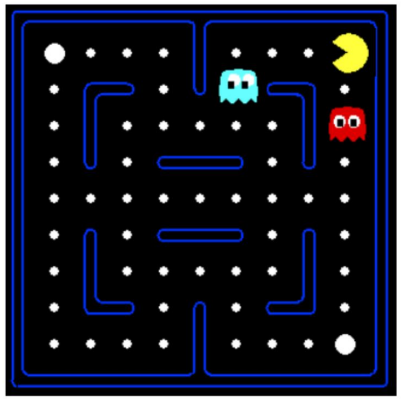


Figure 2

In tabular methods, we know nothing about this state.

Figure 3

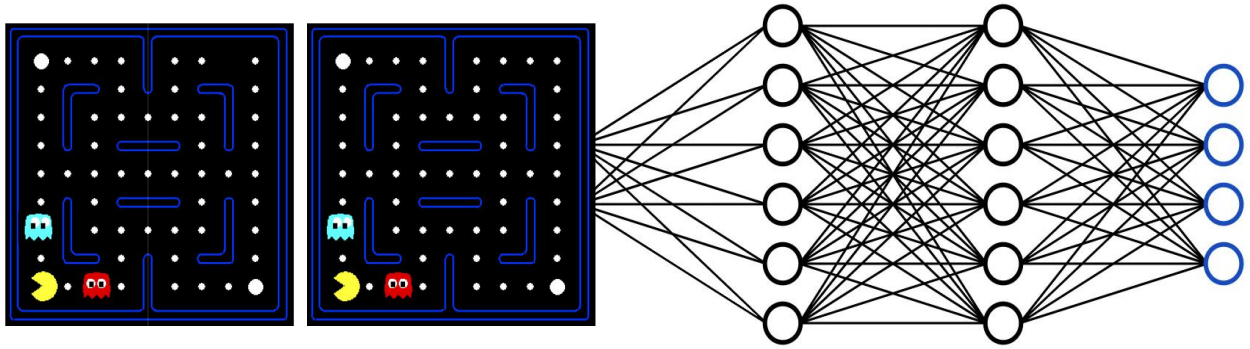
*We know **nothing** about this state either!*

Theorem (Jin et al., NeurIPS 2018). *Without any further structural assumption, the expected regret of any algorithm for episodic MDPs must be at least* $\Omega(\sqrt{SAH^3K})$.

but in fact Figure 1 and 3 are very similar…

Function Approximation

- **Function approximation:** approximate using a parameterized function.
  - Describe states (or state-actions) using feature representations in $\mathbb{R}^d$.
  - A modern choice: DNN as a feature representer



similar output

parameterize MDP model with a low-dimensional representation

⇨ regret bound should not dependent on S or A, but rather the intrinsic dimension d

Function Approximation

$\phi(s'|s,a)$ is the known feature map, and $\theta_h^* \in \mathbb{R}^d$ is unknown parm-meter to estimate

❑ Linear Function Approximation

- Linear mixture MDPs [Ayoub et al., 2020]: $\mathbb{P}_h(s'|s,a) = \phi(s'|s,a)^\top \theta_h^*$
- Linear / low-rank MDPs [Jin et al., 2020]: $\mathbb{P}_h(s'|s,a) = \phi(s,a)^\top \mu^*(s'), r_h(s,a) = \phi(s,a)^\top \theta_h^*$
- …

linearity is hard to satisfy in practice!

Function Approximation

❑ Linear Function Approximation

- Linear mixture MDPs [Ayoub et al., 2020]: $\mathbb{P}_h(s'|s,a) = \phi(s'|s,a)^\top \theta_h^*$
- Linear / low-rank MDPs [Jin et al., 2020]: $\mathbb{P}_h(s'|s,a) = \phi(s,a)^\top \mu^*(s'), r_h(s,a) = \phi(s,a)^\top \theta_h^*$
- …

linearity is hard to satisfy in practice!

Technically, this **"linear"** MDP parametrization arises because it can be reduced to and solved by ***stochastic linear bandits***, which is well-understood.

❑ General Function Approximation

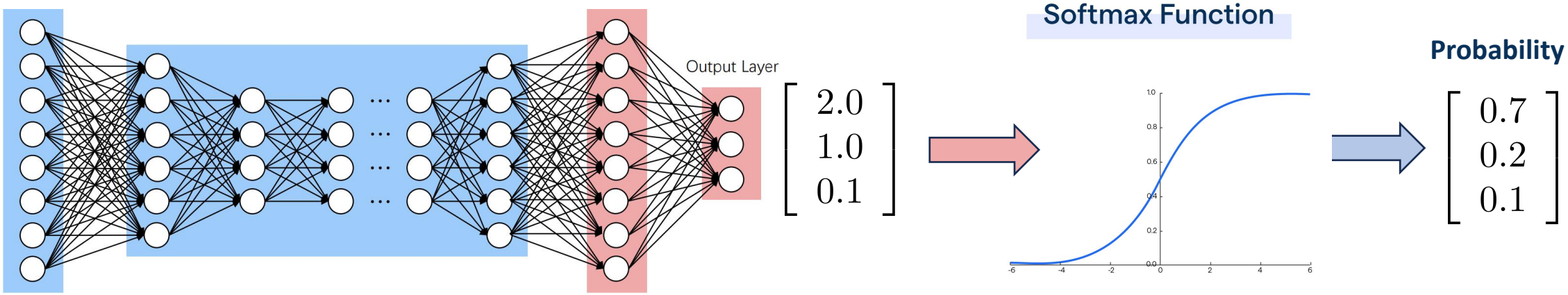
- Eluder dimension [Russo and Roy, 2013, Jin et al., 2021]
- Decision-Estimation Coefficient (DEC) [Foster et al., 2021]
- Admissible Bellman Characterization (ABC) [Chen et al., 2023]
- …

usually no computationally efficient algorithms provided

⇨ ***computationally efficient beyond linearity?***

MNL Function Approximation

- **A new class: Multinomial Logit (MNL) function approximation** [Hwang and Oh, 2023]



MNL mixture MDPs:

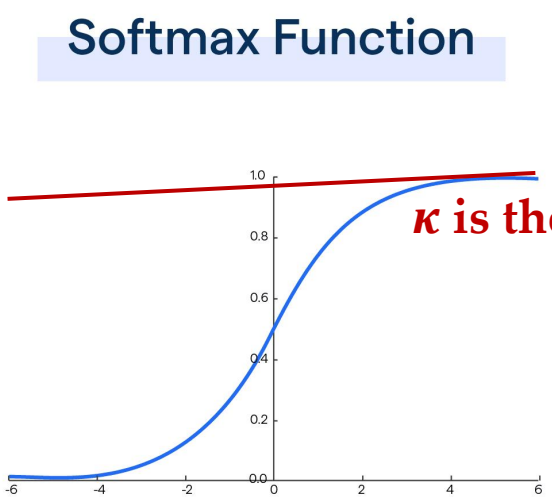
$$\mathbb{P}_h(s' \mid s,a) = \frac{\exp\left(\phi\left(s' \mid s,a\right)^\top \theta_h^*\right)}{\sum_{\widetilde{s}\in\mathcal{S}_{h,s,a}} \exp\left(\phi(\widetilde{s} \mid s,a)^\top \theta_h^*\right)}$$

- $\phi(s'|s,a)$ is the known feature mapping
- $\{\theta_h^*\}_{h=1}^H$ is the unknown transition parameter
- $\mathcal{S}_{h,s,a} \triangleq \{s' \in \mathcal{S} \mid \mathbb{P}_h(s'|s,a) \neq 0\}$ is reachable states

Key Challenge: non-linearity

Linear mixture MDPs: $\mathbb{P}_h(s'|s,a) = \phi(s'|s,a)^\top \theta_h^*$

MNL mixture MDPs: $\mathbb{P}_h(s' \mid s,a) = \dfrac{\exp\left(\phi\left(s' \mid s,a\right)^\top \theta_h^*\right)}{\sum_{\widetilde{s}\in\mathcal{S}_{h,s,a}} \exp\left(\phi(\widetilde{s} \mid s,a)^\top \theta_h^*\right)}$



κ **is the minimum slope**

even two vastly different inputs will have much similar outputs

Regularity assumption:

$$\inf_{\theta\in\Theta} p_{s,a}^{s'}(\theta) p_{s,a}^{s''}(\theta) \geq \kappa$$

where $p_{s,a}^{s'}(\theta) = \dfrac{\exp(\phi(s'|s,a)^\top \theta)}{\sum_{\widetilde{s}\in\mathcal{S}_{s,a}} \exp(\phi(\widetilde{s}|s,a)^\top \theta)}$

Define $U = \max_{(h,s,a)} S_{h,s,a} \Rightarrow \kappa \leq 1/U^2$.

in the worst case, $\kappa^{-1} = \Omega(S^2)$

Main Results

- The ***first*** statistically and computationally efficient algorithm

  Algorithm 1: Independent of κ^{-1} in the dominant term;

  Algorithm 2: based on Algorithm1, further achieve efficient time & storage cost.

- The ***first*** lower bound for this problem

in the worst case, $\kappa^{-1} = \Omega(S^2)$

Reference	Model	Upper Bound	Lower Bound
Zhou et al. [2021]	Linear mixture MDP	$\widetilde{\mathcal{O}}(dH^{3/2}\sqrt{K})$	$\Omega(dH^{3/2}\sqrt{K})$
Hwang and Oh [2023]	MNL mixture MDP	$\widetilde{\mathcal{O}}(\kappa^{-1}dH^2\sqrt{K})$	—
Our work	MNL mixture MDP	$\widetilde{\mathcal{O}}(dH^2\sqrt{K} + \kappa^{-1}d^2H^2)$	$\Omega(dH\sqrt{K})$

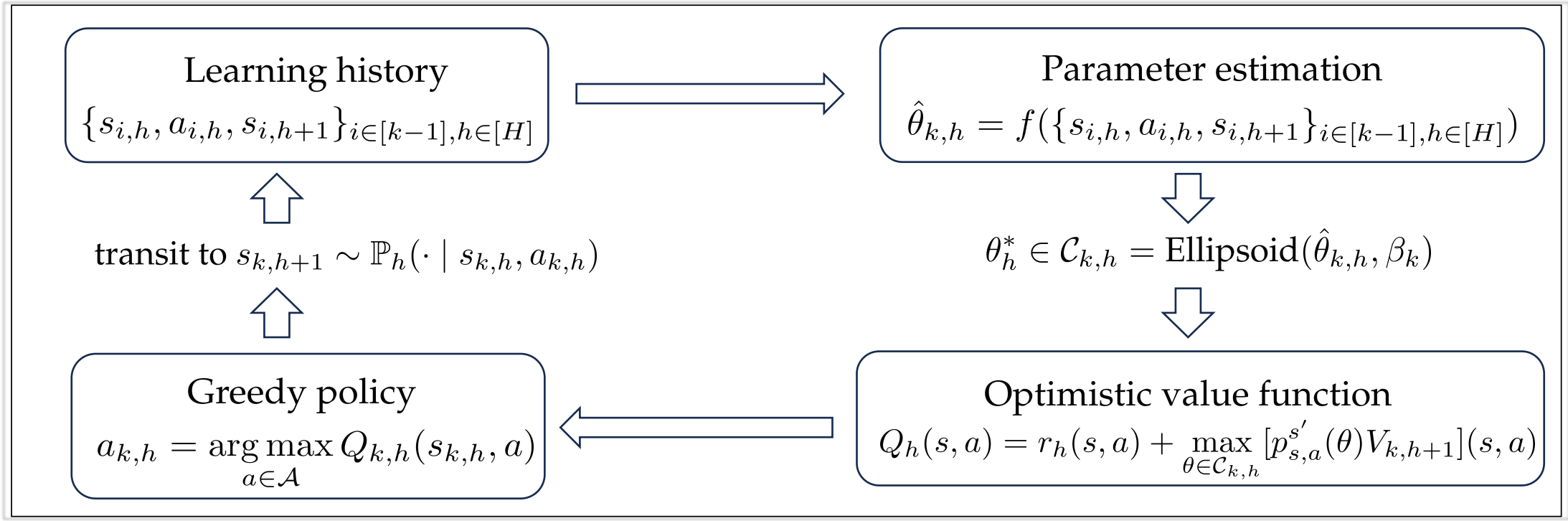
Match the results for linear mixture MDPs except for the dependence on H.

Algorithm: A Pipeline for UCB

(Upper Confidence Bound)

- Parameter estimation
- Confidence region construction
- UCB arm selection

$$\mathbb{P}_h(s' \mid s, a) = \frac{\exp\left(\phi\left(s' \mid s, a\right)^\top \theta_h^*\right)}{\sum_{\widetilde{s} \in \mathcal{S}_{h,s,a}} \exp\left(\phi(\widetilde{s} \mid s, a)^\top \theta_h^*\right)}$$

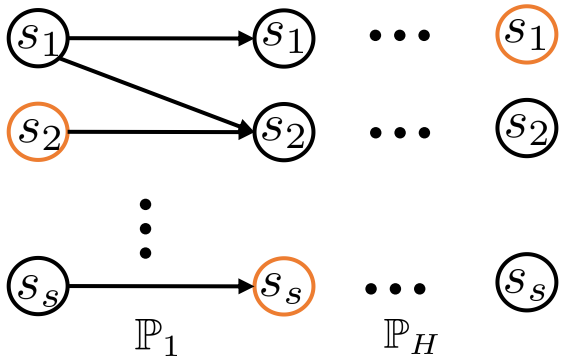


Parameter Estimation

- **Maximum likelihood estimation (MLE)**

$$\widehat{\theta}_{k,h} = \arg\min_{\theta \in \mathbb{R}^d} \left\{ \sum_{i=1}^{k-1} \sum_{s' \in \mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2} \|\theta\|_2^2 \right\} \triangleq \mathcal{L}_{k,h}(\theta)$$

- $y_{k,h}$ is the response variable with $y_{k,h}^{s'} = \mathbb{1}\left(s_{k,h+1} = s'\right)$ for $s' \in \mathcal{S}_{k,h}$.



- Estimation error analysis: with probability at least $1-\delta$, [Hwang and Oh, 2023]

$$\mathbb{P}_h(s' \mid s,a) = \frac{\exp\left(\phi\left(s' \mid s,a\right)^\top \theta_h^*\right)}{\sum_{\widetilde{s} \in \mathcal{S}_{h,s,a}} \exp\left(\phi(\widetilde{s} \mid s,a)^\top \theta_h^*\right)}$$

$$\|\theta_h^* - \widehat{\theta}_{k,h}\|_{A_{k,h}^{-1}} \leq \kappa^{-1} \sqrt{d \log(kH/\delta)}$$

$$A_{k,h} = \sum_{i=1}^{k-1} \sum_{s' \in \mathbb{S}_{i,h}} \phi_{s,a}^{s'} (\phi_{s,a}^{s'})^\top$$

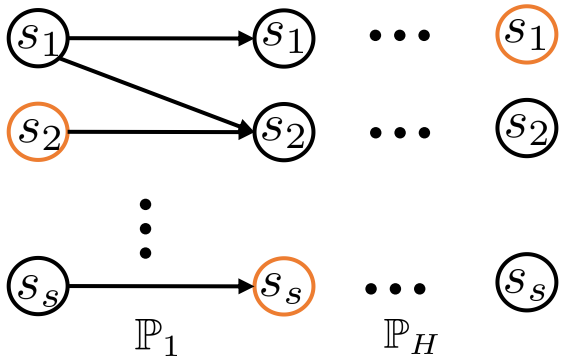
unpleasant dependence on κ^{-1} !

Parameter Estimation

- **Maximum likelihood estimation (MLE)**

$$\widehat{\theta}_{k,h} = \arg\min_{\theta \in \mathbb{R}^d} \left\{ \sum_{i=1}^{k-1} \sum_{s' \in \mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2} \|\theta\|_2^2 \right\} \triangleq \mathcal{L}_{k,h}(\theta)$$

- $y_{k,h}$ is the response variable with $y_{k,h}^{s'} = \mathbb{1}\left(s_{k,h+1} = s'\right)$ for $s' \in \mathcal{S}_{k,h}$.



- Estimation error analysis: with probability at least $1-\delta$, [Li-Zhang-**Z**-Zhou, 2024]

$$\mathcal{G}_{k,h}(\theta) = \sum_{i=1}^{k-1} \sum_{s' \in \mathcal{S}_{i,h}} \left(p_{i,h}^{s'}(\theta) - y_{i,h}^{s'}\right) \phi_{i,h}^{s'} + \lambda_k \theta,$$

$$\mathcal{H}_{k,h}(\theta) = \sum_{i=1}^{k-1} \sum_{s' \in \mathcal{S}_{i,h}} \dot{p}_{i,h}^{s'}(\theta) \phi_{i,h}^{s'} \left(\phi_{i,h}^{s'}\right)^\top + \lambda_k I$$

$$\left\| \mathcal{G}_{k,h}(\theta_h^*) - \mathcal{G}_{k,h}(\widehat{\theta}_{k,h}) \right\|_{\mathcal{H}_{k,h}^{-1}(\theta_h^*)} \leq \sqrt{d \log(kH/\delta)}$$

independent of κ^{-1}*!*

essentially "variance-aware" local norm

Confidence Region + UCB selection

- Confidence region construction

$$\widehat{Q}_{k,h}(s,a) = \left[r_h(s,a) + \max_{\theta \in \widehat{\mathcal{C}}_{k,h}} \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\theta) \widehat{V}_{k,h+1}(s') \right]_{[0,H]}$$

$$\widehat{\mathcal{C}}_{k,h} = \left\{ \theta \in \Theta \mid \left\| \mathcal{G}_{k,h}(\theta) - \mathcal{G}_{k,h}(\widehat{\theta}_{k,h}) \right\|_{\mathcal{H}_{k,h}^{-1}(\theta)} \leq \widehat{\beta}_k \right\}$$

- UCB arm selection

$$\widehat{V}_{k,h}(s) = \arg\max_{a \in \mathcal{A}} \widehat{Q}_{k,h}(s,a)$$

Greedy policy:

$$a_{k,h} = \arg\max_{a \in \mathcal{A}} \hat{Q}_{k,h}(s_{k,h}, a)$$

Theorem 1. *For any $\delta \in (0,1)$, with probability at least $1-\delta$, our algorithm ensures:*

$$\mathrm{Regret}_K \leq \widetilde{\mathcal{O}}\left(dH^2\sqrt{K} + \kappa^{-1} d^2 H^2 \right)$$

The first algorithm with optimal regret without dependence on κ (in dominating term).

Computational Challenge

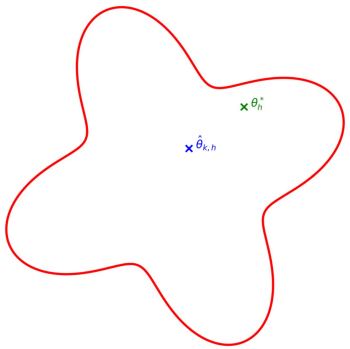
- **MLE estimator**: Computational and storage cost at episode k is $O(k)$!

$$\widehat{\theta}_{k,h} = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_{k,h}(\theta) \triangleq \sum_{i=1}^{k-1} \sum_{s' \in \mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2} \|\theta\|_2^2$$

- **UCB selection**: Feasible domain can be ***non-convex*** !

$$\widehat{Q}_{k,h}(s,a) = \left[r_h(s,a) + \max_{\theta \in \widehat{\mathcal{C}}_{k,h}} \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\theta) \widehat{V}_{k,h+1}(s') \right]_{[0,H]}$$

$$\widehat{\mathcal{C}}_{k,h} = \left\{ \theta \in \Theta \mid \left\| \mathcal{G}_{k,h}(\theta) - \mathcal{G}_{k,h}\left(\widehat{\theta}_{k,h}\right) \right\|_{\mathcal{H}_{k,h}^{-1}(\theta)} \leq \widehat{\beta}_k \right\}$$



Online Mirror Descent

- OMD is a powerful online learning framework to optimize regret.

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \mathcal{D}_\psi(\mathbf{x}, \mathbf{x}_t) \right\}$$

where $\mathcal{D}_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \nabla\psi(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$ is the Bregman divergence.

$$\widetilde{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \left\langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \right\rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_{k,h} \right\|^2_{\widetilde{\mathcal{H}}_{k,h}} \right\}$$

where $\widetilde{\mathcal{H}}_{k,h} = \eta H_{k,h}(\widetilde{\theta}_{k,h}) + \sum_{i=1}^{k-1} H_{i,h}(\widetilde{\theta}_{i+1,h})$

A Summary of OMD Deployment

- Our previous mentioned algorithms can all be covered by OMD.

Algo.	OMD/proximal form	$\psi(\cdot)$	η_t	Regret_T
OGD for convex	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2$	$\|\mathbf{x}\|_2^2$	$\frac{1}{\sqrt{t}}$	$\mathcal{O}(\sqrt{T})$
OGD for strongly c.	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2$	$\|\mathbf{x}\|_2^2$	$\frac{1}{\sigma t}$	$\mathcal{O}(\frac{1}{\sigma}\log T)$
ONS for exp-concave	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_t\|_{A_t}^2$	$\|\mathbf{x}\|_{A_t}^2$	$\frac{1}{\gamma}$	$\mathcal{O}(\frac{d}{\gamma}\log T)$
Hedge for PEA	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\Delta_N} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \text{KL}(\mathbf{x}\|\mathbf{x}_t)$	$\sum_{i=1}^N x_i \log x_i$	$\sqrt{\frac{\ln N}{T}}$	$\mathcal{O}(\sqrt{T\log N})$

Advanced Optimization (Fall 2024) Lecture 6. Online Mirror Descent 64

We here use OMD as a statistical estimation tool!

More details of OMD can be found in Lecture 6 of Advanced Optimization Course 2024 Fall
https://www.pengzhao-ml.com/course/AOpt2024fall/

OMD as Statistical Estimator

- Replace MLE with **Online Mirror Descent (OMD)** inspired by [Zhang and Sugiyama, NeurIPS' 23]

MLE

$$\widehat{\theta}_{k+1,h} = \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{k} \underbrace{\sum_{s'\in\mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2}\|\theta\|_2^2}_{\triangleq \ell_{i,h}(\theta)}$$

one-pass update by OMD ⇩ "lookahead" local norm to keep historical information

***implicit* OMD**

$$\bar{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \ell_{k,h}(\theta) + \frac{1}{2\eta} \left\| \theta - \bar{\theta}_{k,h} \right\|_{\bar{\mathcal{H}}_{k,h}}^2 \right\},$$

where $\bar{\mathcal{H}}_{k,h} \triangleq \sum_{i=1}^{k-1} H_{i,h}\left(\bar{\theta}_{i+1,h}\right)$ is a particular local norm.

Still no closed-form solution!

second-order Taylor expansion ⇩ $\widetilde{\ell}_{k,h}(\theta) = \ell_{k,h}(\widetilde{\theta}_{k,h}) + \langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \rangle + \frac{1}{2}\|\theta - \widetilde{\theta}_{k,h}\|_{H_{k,h}(\widetilde{\theta}_{k,h})}^2$

standard OMD

$$\widetilde{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \left\langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \right\rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_{k,h} \right\|_{\widetilde{\mathcal{H}}_{k,h}}^2 \right\},$$

where $\widetilde{\mathcal{H}}_{k,h} = \eta H_{k,h}(\widetilde{\theta}_{k,h}) + \sum_{i=1}^{k-1} H_{i,h}(\widetilde{\theta}_{i+1,h})$ incoporates additional second-order quantity.

OMD: Estimation Error

$$\widetilde{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \left\langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \right\rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_{k,h} \right\|^2_{\widetilde{\mathcal{H}}_{k,h}} \right\},$$

where $\widetilde{\mathcal{H}}_{k,h} = \eta H_{k,h}(\widetilde{\theta}_{k,h}) + \sum_{i=1}^{k-1} H_{i,h}(\widetilde{\theta}_{i+1,h})$ incoporates additional second-order quantity.

can be solved with two steps:

$$\widetilde{\theta}'_{k+1,h} = \widetilde{\theta}_{k,h} - \eta \widetilde{\mathcal{H}}^{-1}_{k,h} \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}),$$

$$\widetilde{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\| \theta - \widetilde{\theta}'_{k+1,h} \right\|^2_{\widetilde{\mathcal{H}}_{k,h}}$$

- ✓ *Hessian can be analytically calculated*
- ✓ *inversion can be easily computed by rank-1 update, $O(d^2)$ complexity*

Lemma (Confidence Set). *For any $\delta \in (0,1)$, set $\eta = \frac{1}{2}\log(1+U) + (B+1)$ and $\lambda = 84\sqrt{2}\eta(B+d)$, define*

$$\widetilde{\mathcal{C}}_{k,h} = \left\{ \theta \in \Theta \mid \left\| \theta - \widetilde{\theta}_{k,h} \right\|_{\mathcal{H}_{k,h}} \le \mathcal{O}(\sqrt{d}\log U \log(kH/\delta)) \triangleq \widetilde{\beta}_k \right\}.$$

Then, we have $\Pr[\theta^*_h \in \widetilde{\mathcal{C}}_{k,h}] \ge 1-\delta, \forall k \in [K], h \in [H]$.

independent of κ^{-1}!

Key Analysis

A general template of OMD estimator:

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \left\{ g_t(\theta) + \frac{1}{2\eta} \|\theta - \theta_t\|_{A_t}^2 \right\}$$

where $g_t(\theta)$ is the surrogate loss and A_t is the local norm.

Lemma 1. *For OMD estimator, we have*

$$\frac{1}{2\eta} \|\theta_{t+1} - \theta_*\|_{A_t}^2 \leq \langle \nabla g_t(\theta_t), \theta_t - \theta_* \rangle + \frac{1}{2\eta} \|\theta_t - \theta_*\|_{A_t}^2 - \frac{1}{2\eta} \|\theta_{t+1} - \theta_t\|_{A_t}^2.$$

A proper choice of the local norm A_t and the surrogate loss $g_t(\theta)$ become highly crucial.

⇨ Self-concordance of logistic loss & Second-order approximation & Negative regret in OMD

Computational Challenge

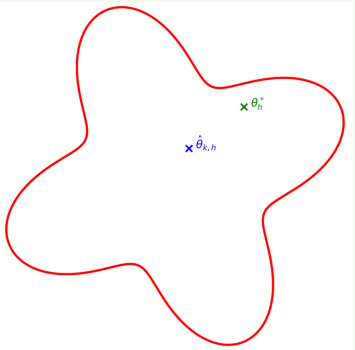
- **MLE estimator**: Computational and storage cost at episode k is $O(k)$!

$$\widehat{\theta}_{k,h} = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_{k,h}(\theta) \triangleq \sum_{i=1}^{k-1} \sum_{s' \in \mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2} \|\theta\|_2^2$$

- **UCB selection**: Feasible domain can be ***non-convex*** !

$$\widehat{Q}_{k,h}(s,a) = \left[r_h(s,a) + \max_{\theta \in \widehat{\mathcal{C}}_{k,h}} \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\theta) \widehat{V}_{k,h+1}(s') \right]_{[0,H]}$$

$$\widehat{\mathcal{C}}_{k,h} = \left\{ \theta \in \Theta \mid \left\| \mathcal{G}_{k,h}(\theta) - \mathcal{G}_{k,h}\left(\widehat{\theta}_{k,h}\right) \right\|_{\mathcal{H}_{k,h}^{-1}(\theta)} \leq \widehat{\beta}_k \right\}$$



Efficient Optimistic Value Function

$$\widehat{Q}_{k,h}(s,a) = \left[r_h(s,a) + \max_{\theta \in \widehat{\mathcal{C}}_{k,h}} \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\theta) \widehat{V}_{k,h+1}(s') \right]_{[0,H]}, \quad \widehat{\mathcal{C}}_{k,h} = \left\{ \theta \in \Theta \mid \left\| \mathcal{G}_{k,h}(\theta) - \mathcal{G}_{k,h}\left(\widehat{\theta}_{k,h}\right) \right\|_{\mathcal{H}_{k,h}^{-1}(\theta)} \leq \widehat{\beta}_k \right\}$$

non-convex

- Replace maximization with closed-form bonus:

$$\widetilde{Q}_{k,h}(s,a) = \left[r_h(s,a) + \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\widetilde{\theta}_{k,h}) \widetilde{V}_{k,h+1}(s') + \underline{\epsilon_{s,a}^{\text{fst}} + \epsilon_{s,a}^{\text{snd}}} \right]_{[0,H]} \qquad \widetilde{V}_{k,h}(s) = \arg\max_{a \in \mathcal{A}} \widetilde{Q}_{k,h}(s,a)$$

Preserve local information effectively!

bound the value difference by second-order Taylor expansion

Lemma. *For any $V : \mathcal{S} \to [0, H]$ and $(h, s, a) \in [H] \times \mathcal{S} \times \mathcal{A}$, it holds*

$$\left| \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\widetilde{\theta}_{k,h}) V(s') - \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\theta_h^*) V(s') \right| \leq \epsilon_{s,a}^{\text{fst}} + \epsilon_{s,a}^{\text{snd}}$$

where

$$\epsilon_{s,a}^{\text{fst}} = H\widetilde{\beta}_k \sum_{s' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s'}(\widetilde{\theta}_{k,h}) \left\| \phi_{s,a}^{s'} - \sum_{s'' \in \mathcal{S}_{h,s,a}} p_{s,a}^{s''}(\widetilde{\theta}_{k,h}) \phi_{s,a}^{s''} \right\|_{\mathcal{H}_{k,h}^{-1}}, \epsilon_{s,a}^{\text{snd}} = \frac{5}{2} H \widetilde{\beta}_k^2 \max_{s' \in \mathcal{S}_{h,s,a}} \left\| \phi_{s,a}^{s'} \right\|_{\mathcal{H}_{k,h}^{-1}}^2$$

Algorithm and Regret Bound

Algorithm 2 UCRL-MNL-LL

Input: Step size η, regularization parameter λ, confidence width $\widetilde{\beta}_k$, confidence parameter δ.

Initialization: $\mathcal{H}_{1,h} = \lambda I, \widehat{\theta}_{1,h} = \mathbf{0}$ for all $h \in [H]$.
for $k = 1, \ldots, K$ **do**
  Compute $\widetilde{Q}_{k,h}(\cdot,\cdot)$ in a backward way as in (11).
  for $h = 1, \ldots, H$ **do**
    Observe state $s_{k,h}$, select action $a_{k,h} = \arg\max_{a \in \mathcal{A}} \widetilde{Q}_{k,h}(s_{k,h}, a)$.
    Update $\widetilde{\mathcal{H}}_{k,h} = \mathcal{H}_{k,h} + \eta H_{k,h}(\widetilde{\theta}_{k,h})$.
    Compute $\widetilde{\theta}_{k+1,h} = \arg\min_{\theta \in \Theta} \langle g_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \rangle + \frac{1}{2\eta} \|\theta - \widetilde{\theta}_{k,h}\|_{\widetilde{\mathcal{H}}_{k,h}}$.
    Update $\mathcal{H}_{k,h} = \mathcal{H}_{k,h} + H_{k,h}(\widetilde{\theta}_{k+1,h})$.
  end for
end for

$$\widetilde{Q}_{k,h}(s,a) = \left[r_h(s,a) + \sum_{s' \in \mathcal{S}_{h,s,a}} p^{s'}_{s,a}(\widetilde{\theta}_{k,h}) \widetilde{V}_{k,h+1}(s') + \epsilon^{\text{fst}}_{s,a} + \epsilon^{\text{snd}}_{s,a} \right]_{[0,H]}$$

Computational complexity:

$O(A)$
$O(U^2 d^2)$
$O(d^3)$
$O(U^2 d^2)$

Total: $(d^3 + U^2 d^2 + A) \cdot H \cdot K$

Theorem 2. *For any $\delta \in (0,1)$, with probability at least $1-\delta$, our algortihm ensures:*

$$\text{Regret}_K \leq \widetilde{\mathcal{O}}\left(dH^2\sqrt{K} + \kappa^{-1} d^2 H^2 \right)$$

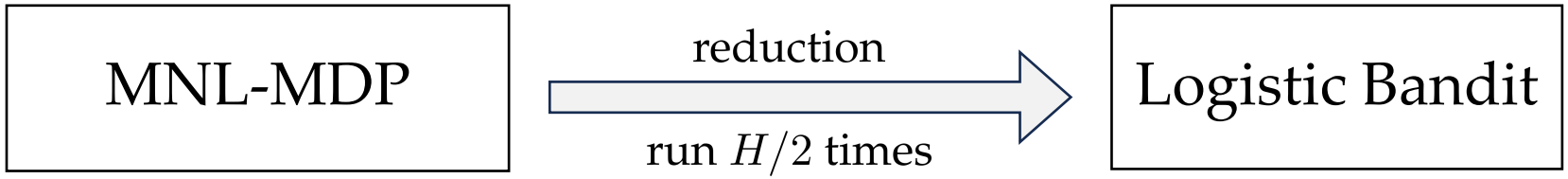
κ^{-1}-independent regret (in main term) & constant computational cost per round

Lower bound for MNL-MDP

- Lower bound by reducing MNL-MDP as Logistic Bandits.

Logistic Bandit Problem

- At each round $k \in [K]$, learner selects an action $a_k \in \mathcal{A} \in \mathbb{R}^d$, receives a reward r_k sampled from Bernoulli distribution with $\mu\left(a_k^\top \theta^*\right) = \left(1 + \exp\left(-a_k^\top \theta^*\right)\right)^{-1}$.
- Learner aims to minimize: $\text{Regret}_K^{\text{LB}} = \max_{a \in \mathcal{A}} \sum_{k=1}^K \mu\left(a^\top \theta^*\right) - \sum_{k=1}^K \mu\left(a_k^\top \theta^*\right)$.



Theorem 3. *The lower bound of MNL MDPs with infinite actions is* $\text{Regret}_K \geq \Omega\left(dH\sqrt{K}\right)$.

Note: The follow-up work of [Park et al., AISTATS 2025] improves the lower bound to $\Omega\left(d\, H^{3/2}\sqrt{K}\right)$.

Summary of Online MDPs

- **Upper bound side: avoid dependence on $\kappa^{-1} = \Omega(S^2)$**
  - Avoid MLE estimator: OMD with a suitable local norm
  - Avoid non-convex issue in Q: second-order value difference

Theorem 2. *For any $\delta \in (0,1)$, with probability at least $1-\delta$, our algortihm ensures:*

$$\mathrm{Regret}_K \leq \widetilde{\mathcal{O}}\left(dH^2\sqrt{K} + \kappa^{-1}d^2H^2\right)$$

κ^{-1}-independent regret (in main term) & constant computational cost per round

- **Lower bound side:** reduction to a logistic bandits

Theorem 3. *The lower bound of MNL MDPs with infinite actions is* $\mathrm{Regret}_K \geq \Omega\left(dH\sqrt{K}\right)$.

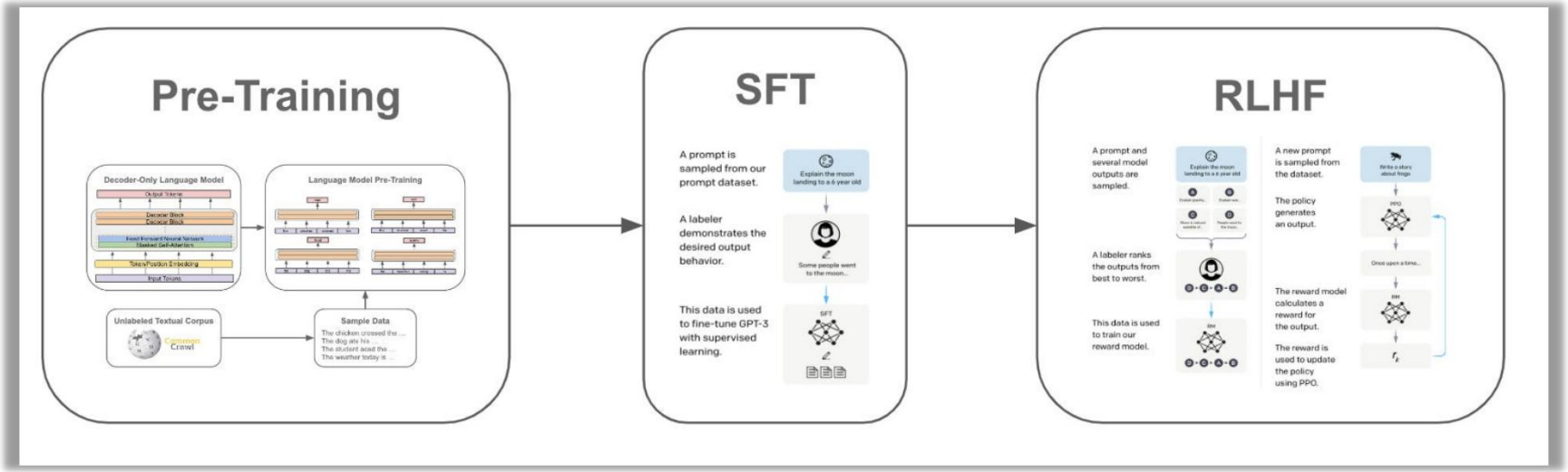
Note: The follow-up work of [Park et al., AISTATS 2025] improves the lower bound to $\Omega\left(d\,H^{3/2}\sqrt{K}\right)$.

Outline

- RL Background
- RL with Function Approximation
- RL with Human Feedback
- Conclusion

Large Language Models

- Three typical stages of LLM training



- **Pre-Training**: Train on large-scale, diverse datasets to learn general capabilities.
- **SFT**: fine-tune the model using labeled data to improve ability to follow instructions.
- **RLHF *(or preference optimization)*** : align model towards human preferences or values.

RLHF for Alignment

- **Input:** a preference 4-argument tuple (x, a, a', y)
  - x: the prompt: `"Please write a joke for me."`
  - a: the first response: `"Sorry, I can't."`
  - a': the second response: `"Here is a joke for you: …"`
  - $y \in \{0, 1\}$: the label (human's preference): a'

- RLHF wants to use input to improve LLM

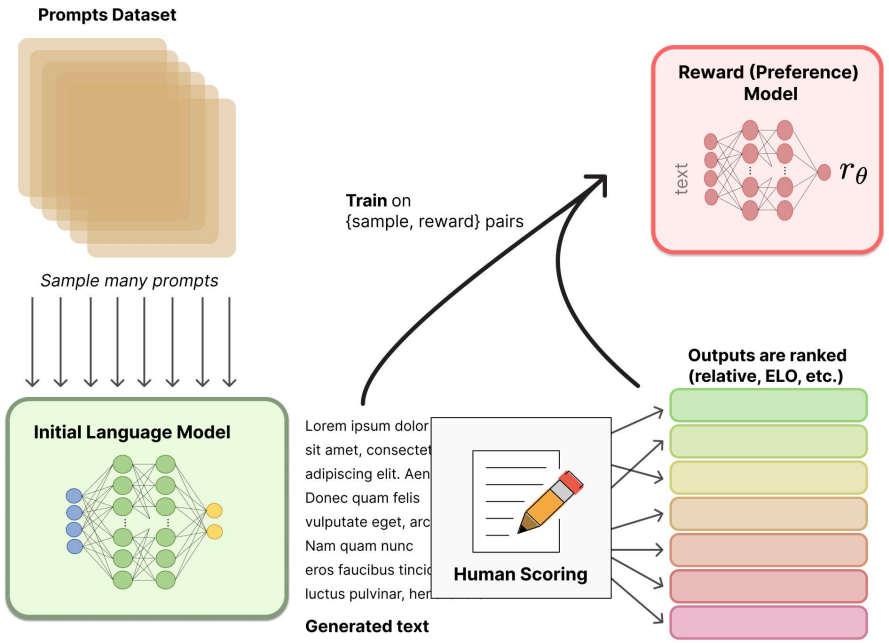
  i.e., *align LLM with human's preference or value (encoded in the preference data)*

- **Output:** a fine-tuned LLM with better aligned preference

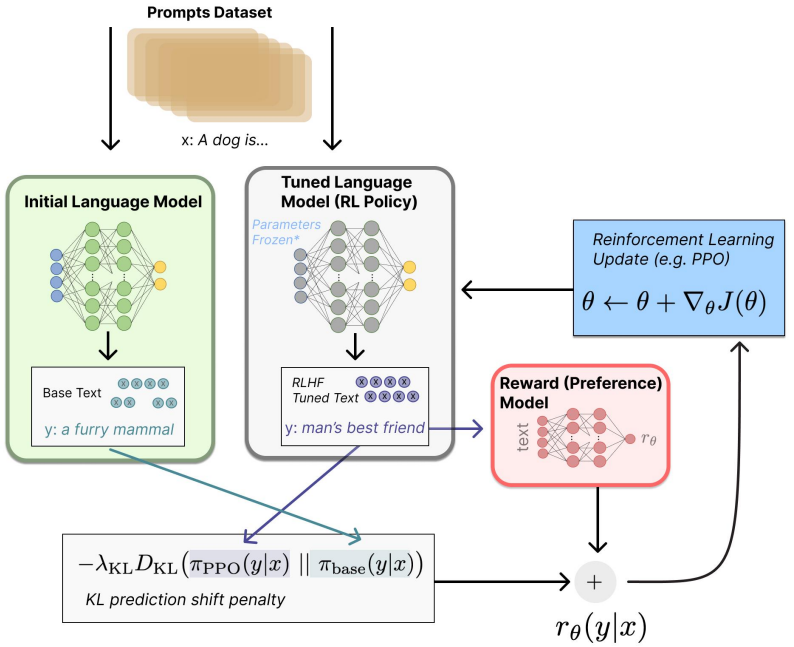
RLHF for Alignment

- A standard pipeline of RLHF

(i) reward model learning



(ii) policy optimization (guided by reward model)



Reward model learning

- How to model the underlying reward based on observed data?

Definition 1 (Bradley-Terry Model). Given a context $x \in \mathcal{X}$ and two actions $a, a' \in \mathcal{A}$, the probability of the human preferring action a over action a' is given by

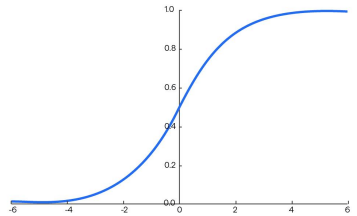
$$\mathbb{P}\left(a \succ a' \mid x\right) = \frac{\exp\left(r\left(x, a\right)\right)}{\exp\left(r\left(x, a\right)\right) + \exp\left(r\left(x, a'\right)\right)}$$

where r is the latent function.

- **Maximum Likelihood Estimation (MLE)**

$$\arg\min_{r_\phi} \mathcal{L}_R\left(r_\phi, \mathcal{D}\right) = -\mathbb{E}_{(x, a_w, a_l) \sim \mathcal{D}}\left[\log \sigma\left(r_\phi(x, a_w) - r_\phi(x, a_l)\right)\right]$$

$$\sigma(w) = \frac{1}{1 + e^{-w}}$$



Connection to MNL mixture MDPs

- **Linear reward model assumption**

$$r(x,a) = \phi(x,a)^\top \theta^*$$

Bradley-Terry Model

$$\mathbb{P}\left(a \succ a' \mid x\right) = \frac{\exp\left(\phi(x,a)^\top \theta^*\right)}{\exp\left(\phi(x,a)^\top \theta^*\right) + \exp\left(\phi(x,a')^\top \theta^*\right)}$$

- MLE estimator

$$r = \arg\min_{r_\phi} -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi\left(x,y_w\right) - r_\phi\left(x,y_l\right)\right)\right]$$

Compared to MNL mixture MDPs

- **Transition Model**

$$\mathbb{P}_h(s' \mid s,a) = \frac{\exp\left(\phi\left(s' \mid s,a\right)^\top \theta_h^*\right)}{\sum_{\widetilde{s}\in\mathcal{S}_{h,s,a}} \exp\left(\phi(\widetilde{s} \mid s,a)^\top \theta_h^*\right)}$$

- **MLE**

$$\theta_{k,h} = \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{k-1} \sum_{s'\in\mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda}{2}\|\theta\|_2^2$$

$$\kappa = \min_{x\in\mathcal{X},a,a'\in\mathcal{A}} \min_{\theta\in\Theta} \dot{\sigma}\left(\phi(x,a)^\top\theta - \phi\left(x,a'\right)^\top\theta\right)$$

"non-linearity coefficient"

We can apply OMD to improve the computational and sample efficiency!

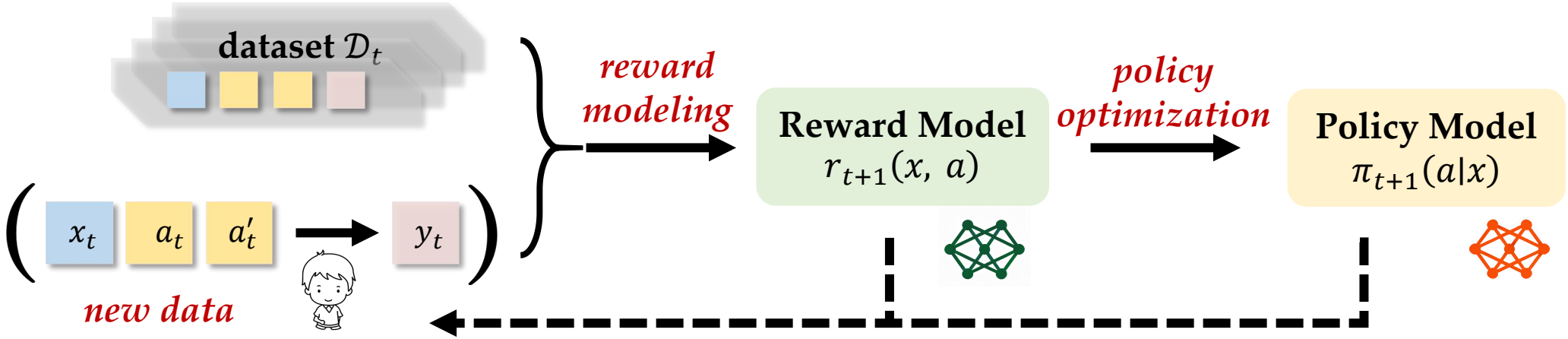
Online RLHF

General Framework of Online RLHF

1 : **New data collection**: sample a tuple $(x_t,\ a_t,\ a'_t)$, obtain the preference label y_t, expand the dataset: $\mathcal{D}_{t+1} = \mathcal{D}_t \cup (x_t,\ a_t,\ a'_t,\ y_t)$

2 : **Reward Modeling**: Train reward model r_{t+1} based on dataset $\mathcal{D}_{t+1}$

3 : **Policy Optimization**: Update the policy π_{t+1} using the learned reward model r_{t+1}



Online RLHF

General Framework of Online RLHF

1: **New data collection**: sample a tuple (x_t, a_t, a_t'), obtain the preference label y_t, expand the dataset: $\mathcal{D}_{t+1} = \mathcal{D}_t \cup (x_t, a_t, a_t', y_t)$

2: **Reward Modeling**: Train reward model r_{t+1} based on dataset $\mathcal{D}_{t+1}$

3: **Policy Optimization**: Update the policy π_{t+1} using the learned reward model r_{t+1}

Reward Modeling: Maximum Likelihood Estimation (MLE)

Define feature difference: $z_t = \phi(x_t, a_t) - \phi(x_t, a_t')$

$$\widehat{\theta}_{t+1} = \arg\min_{\theta \in \mathbb{R}^d} \sum_{s=1}^{t} \ell_s(\theta),$$

where $\ell_t(\theta) = -y_t \log\left(\sigma\left(z_t^\top \theta\right)\right) - (1 - y_t) \log\left(1 - \sigma\left(z_t^\top \theta\right)\right)$

At iteration t:
time complexity: $O(t \log t)$,
storage complexity: $O(t)$

One-Pass Estimation

- Online Mirror Descent

Define gradient and Hessian: $g_t(\theta) = \left(\sigma\left(z_t^\top \theta\right) - y_t\right) z_t, \quad H_t(\theta) = \dot{\sigma}\left(z_t^\top \theta\right) z_t z_t^\top$

$$\widetilde{\theta}_{t+1} = \arg\min_{\theta \in \Theta} \left\{ \langle g_t(\widetilde{\theta}_t), \theta \rangle + \frac{1}{2\eta} \|\theta - \widetilde{\theta}_t\|_{\widetilde{\mathcal{H}}_t}^2 \right\}, \quad \text{where } \widetilde{\mathcal{H}}_t = \sum_{i=1}^{t-1} H_i(\widetilde{\theta}_{i+1}) + \eta H_t(\widetilde{\theta}_t) + \lambda I.$$

constant time and storage complexity, independent of t

look-ahead local norm

second-order approximation

- Estimation Error analysis:

$$\left\|\theta - \widetilde{\theta}_t\right\|_{\mathcal{H}_t} \leq \mathcal{O}\left(\sqrt{d}(\log(t/\delta))^2\right)$$

Enjoy (basically) the same order estimation error guarantee as MLE!

On-Policy Data Collection

❑ Case 1: on-policy data collection

- **Data collection:** uniform sampling $x_t \sim \rho$, and $a_t, a_t' \sim \pi_t(\cdot|x_t)$
- **Policy Optimization:** $\pi_t = \arg\max_\pi \widetilde{J}_t(\pi)$,

where $\widetilde{J}_t(\pi) = \mathbb{E}_{x\sim\rho, a\sim\pi(\cdot|x)}\left[\phi(x,a)^\top \widetilde{\theta}\right] + \widetilde{\beta}_t \left\|\mathbb{E}_{\mathbf{x}\sim\rho}[x, \pi(x)]\right\|_{\tilde{\mathcal{H}}_t}$ *exploration bonus*

$$\begin{aligned}\mathrm{SubOpt}(\pi_t) &= \mathbb{E}_{x\sim\rho}\left[r\left(x, \pi^*(x)\right) - r(x, \pi_t(x))\right] \\ &\leq \widetilde{\mathcal{O}}\left(\sqrt{d} \cdot \|\mathbb{E}_{x\sim\rho}[\phi\left(x, \pi_t(x)\right)]\|_{\mathcal{H}_t^{-1}}\right)\end{aligned}$$

"concentratability coefficient" can be bounded by self-normalized concentration

$$\mathrm{Regret}_T = \sum_{t=1}^{T}\left[J\left(\pi^\star\right) - J\left(\pi_t\right)\right] \leq \widetilde{\mathcal{O}}\left(d\sqrt{\frac{T}{\kappa}}\right)$$

Active Data Collection

❑ Case 2: active data collection

- **Data collection:** maximum uncertainty

$$(x_t, a_t, a_t') = \underset{x,a,a' \in \mathcal{X}\times\mathcal{A}\times\mathcal{A}}{\arg\max} \left\{ \|\phi(x,a) - \phi(x,a')\|_{\mathcal{H}_t^{-1}} \right\}$$

- **Policy Optimization**

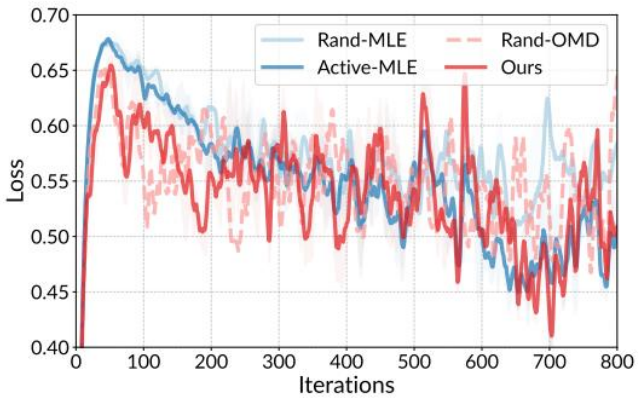
$$\pi_T(x) = \arg\max_a \widetilde{r}_T(x,a) \text{ where } \widetilde{r}_T(x,a) = \frac{1}{T}\sum_{t=1}^{T} \phi(x,a)\tilde{\theta}_t$$

No need to additional exploration due the active data collection strategy

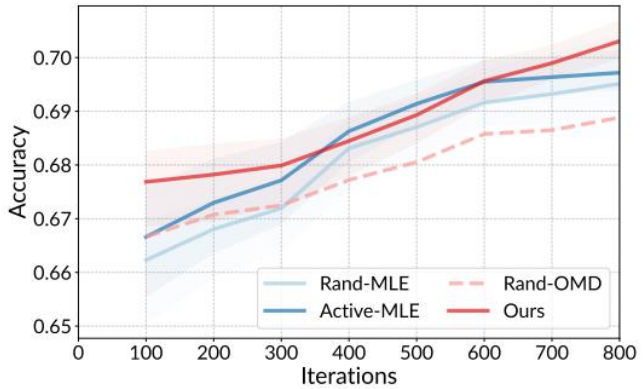
$$\mathrm{SubOpt}(\pi_T) = \mathbb{E}_{x\sim\rho}\left[r\left(x, \pi^*(x)\right) - r(x, \pi_T(x))\right] \leq \widetilde{\mathcal{O}}\left(d\sqrt{\frac{1}{\kappa T}}\right)$$

Experiment

- **Train *Llama-3-8B-Instruct* on *Ultrafeedback* dataset**
- Contenders:
  - (1) On-policy (rand) + MLE (2) Active + MLE (3) On-policy (rand) + MLE (4) Active + OMD



(a) training loss



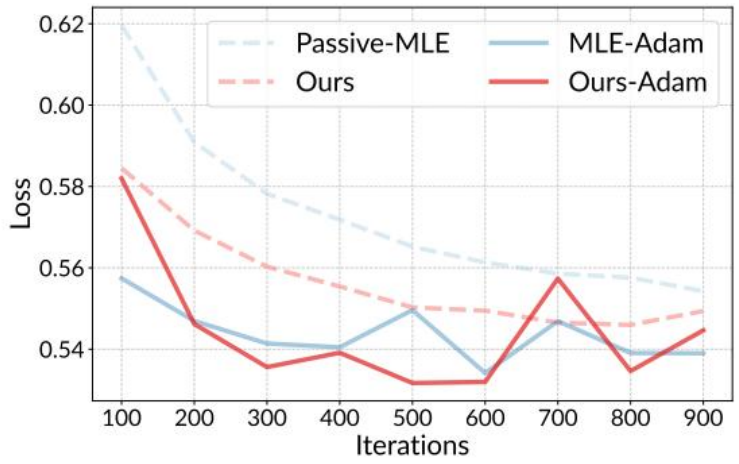
(b) evaluation accuracy

Method	ACC (%)	Time (s)
Rand-MLE	69.51±0.5	4876±47
Active-MLE	69.82±0.4	4982±52
Rand-OMD	68.97±0.6	**1456±31**
Active-OMD	70.43±0.3	**1489±36**

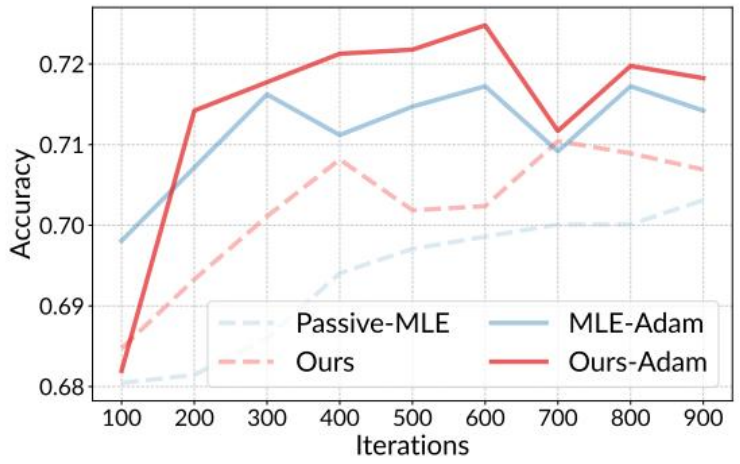
Our OMD-based estimator achieves performance comparable to MLE while offering significantly better computational efficiency.

Experiment

- Interesting point: combined with Adam Optimizer
- Contenders:
  - (1) MLE + SGD (2) MLE + Adam (3) OMD + SGD (4) OMD + Adam



(c) evaluation loss



(d) evaluation accuracy

Adam itself may have already capture some "local" second-order information.

Our OMD-based estimator can be combined with Adam optimizer to further boost the performance.

Outline

- RL Background
- RL with Function Approximation
- RL with Human Feedback
- Conclusion

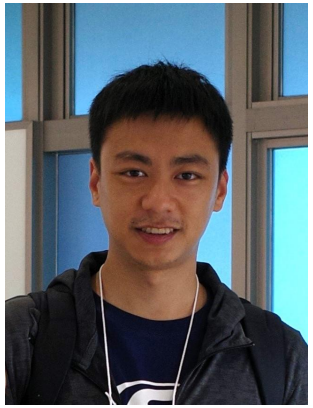
Conclusion

- **Provable efficiency in online RL**
  - Online Mirror Descent (OMD) as a statistical estimator with ***one-pass*** update
- **Multinomial Logit (MNL) Mixture function approximation**
  - A new logistic model to capture the non-linearity of the transition matrix
  - MLE estimator: analysis exploiting local information is vital for statistical efficiency
  - OMD estimator: designing special local norm to replace MLE, keep regret optimality and meanwhile achieve the "one-pass" computational efficiency
- **RL with human feedback**
  - BT model naturally involves the logistic kind non-linearity
  - OMD estimator: used to replace MLE offline estimator, encouraging results

Thanks!

Joint work with

Long-Fei Li
(NJU→ Noah's Ark Lab)



Yu-Jie Zhang
(NJU → U Tokyo)

Yu-Yang Qian (NJU)

Zhi-Hua Zhou (NJU)

- Long-Fei Li, Yu-Jie Zhang, Peng Zhao, and Zhi-Hua Zhou. Provably Efficient Reinforcement Learning with Multinomial Logit Function Approximation. NeurIPS 2024.
- Long-Fei Li*, Yu-Yang Qian*, Peng Zhao, and Zhi-Hua Zhou. Provably Efficient RLHF Pipeline: A Unified View from Contextual Bandits. ArXiv preprint: 2502.07193, 2025

Thanks!

Other References

- Y.-J. Zhang and M. Sugiyama. Online (Multinomial) Logistic Bandit: Improved Regret and Constant Computation Cost. NeurIPS 2023.
- Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved Algorithms for Linear Stochastic Bandits. NIPS 2011.
- L. Faury, M. Abeille, K.-S. Jun, and C. Calauzènes. Jointly Efficient and Optimal Algorithms for Logistic Bandits. AISTATS 2022.
- C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably Efficient Reinforcement Learning with Linear Function Approximation. COLT 2020.
- A. Ayoub, Z. Jia, C. Szepesvári, M. Wang, and L. F. Yang. Model-Based Reinforcement Learning with Value-Targeted Regression. ICML 2020.
- C. Jin, Q. Liu, and S. Miryoosef. Bellman Eluder Dimension: New Rich Classes of RL Problems, and Sample-Efficient Algorithms. NeurIPS 2021.
- D. J. Foster, A. Rakhlin, A. Sekhari, and K. Sridharan. On the Complexity of Adversarial Decision Making. NeurIPS 2022.
- Z. Chen, C. J. Li, H. Yuan, Q. Gu, and M. Jordan. A General Framework for Sample-Efficient Function Approximation in Reinforcement Learning. ICLR 2023.