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

LAMDA
Learning And Mining from DatA

One-Pass Bandit Learning for RLHF and Function Approximation

Peng Zhao
School of AI
Nanjing University

Nov 23, 2025 @ CFAI

Outline

- Bandits Problem
- One-Pass Bandits
- RL Implications
- Summary

Outline

- Bandits Problem
- One-Pass Bandits
- RL Implications
- Summary

Bandits: Interactive Learning

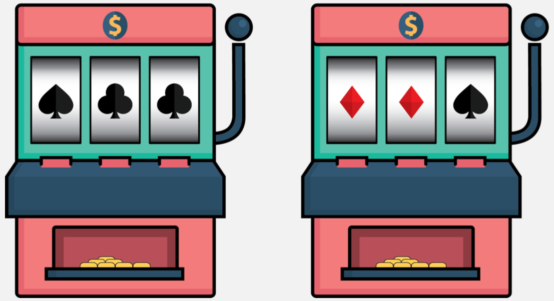
❑ Multi-armed bandits: a simplest formulation for bandit problems

At each round $t = 1, 2, \cdots$

(1) player first chooses an arm $a_t \in [K]$;

(2) environment reveals a reward $r_t(a_t) \sim$ distribution $\mathcal{D}_{a_t}$;

(3) player updates the strategy by the pair $(a_t, r_t(a_t))$.



The goal is to minimize the ***regret*** :

$$\mathbf{Reg}_T \triangleq \max_{a \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} r_t(a) - \sum_{t=1}^{T} r_t(a_t)\right]$$

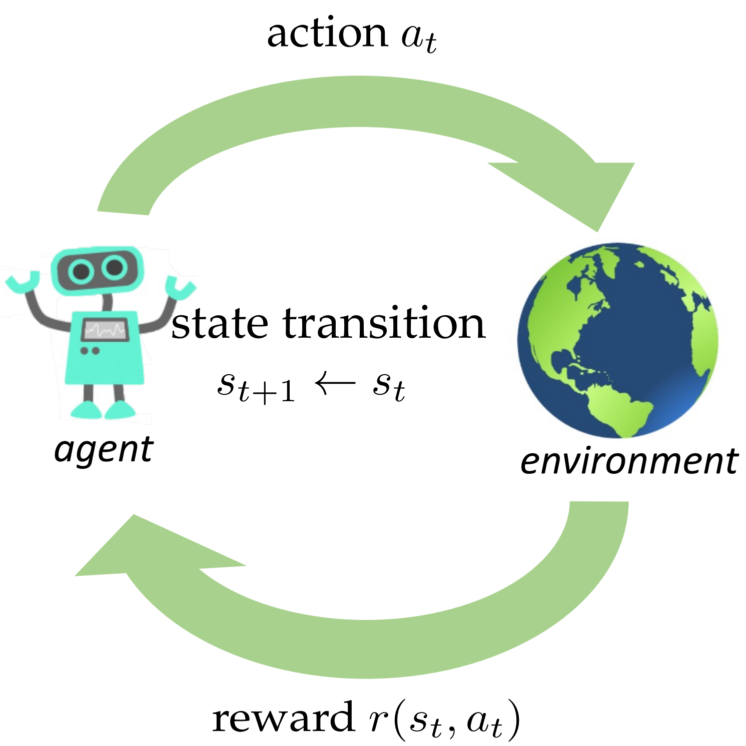
Exploration-Exploitation tradeoff

- **Exploitation**: pull the best arm so far
- **Exploration**: try other arms that may be better

i.e., difference between the cumulative reward of the best arm and that obtained by the bandit algorithm

Bandits: Interactive Learning

- Bandit is ***"single-step"*** decision version of Reinforcement Learning

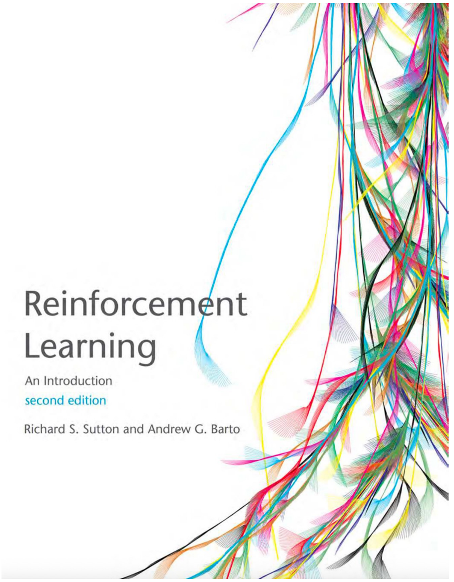


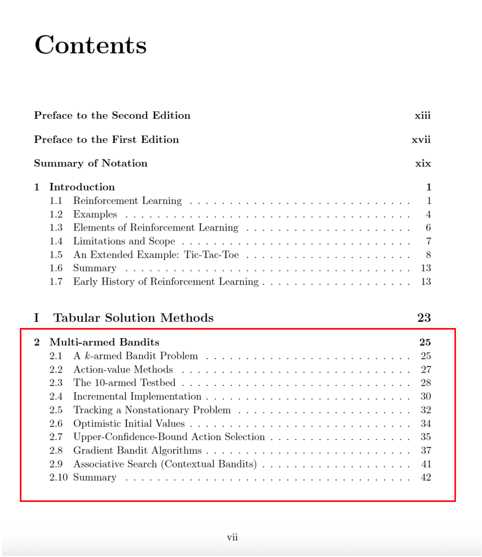
Reinforcement learning:

- Sequential decision making
- With state transition

Bandits:

- Single-step decision making
- No state transition





Contents

vii

Sutton & Barto. Reinforcement Learning, second edition: An Introduction. MIT Press, 2018.

Linear Bandits: Context Matters

❑ Linear Bandits:

$$r_t(x) = x^\top \theta_* + \eta_t$$

- each arm is with a *feature (context)* vector x

– for some unknown parameter θ_*;

– with unknonw noise: η_t is sub-Gaussian noise

- Regret measure: $\bar{R}_T \triangleq \sum_{t=1}^{T} \max_{\mathbf{x} \in \mathcal{X}_t} \mathbf{x}^\top \theta_* - \sum_{t=1}^{T} X_t^\top \theta_*$

Example: ***book recommendation***

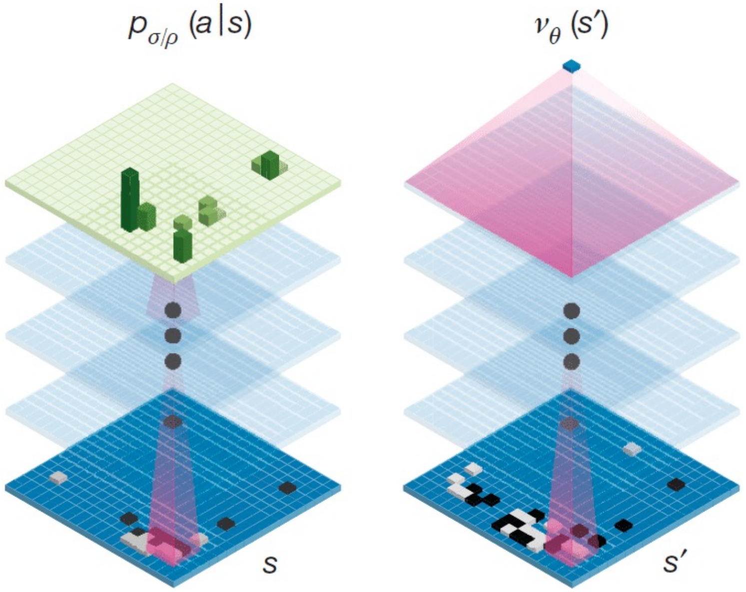
- Each arm is a book with side information;
- Arm set could be very large or even infinite.

- LinUCB [Abbasi-Yadkori et al., NIPS'11] : first estimate the parameter, then construct UCB to select arm

Linear bandit serves as the most basic structural bandit problem, also acts as the fundamental tool to analyze RL/control theory, particularly about function approximation

Linear bandits for RL Theory

Function Approximation



a technique with huge success (especially by involving DNN) , crucially useful for the AlphaGo's success

Provably Efficient Reinforcement Learning with Linear Function Approximation

Chi Jin
University of California, Berkeley
chijin@cs.berkeley.edu

Zhuoran Yang
Princeton University
zy6@princeton.edu

Zhaoran Wang
Northwestern University
zhaoranwang@gmail.com

Michael I. Jordan
University of California, Berkeley
jordan@cs.berkeley.edu

COLT 2020

Reinforcement Learning in Feature Space: Matrix Bandit, Kernels, and Regret Bound

Lin F. Yang
Princeton University
lin.yang@princeton.edu

Mengdi Wang
Princeton University
mengdiw@princeton.edu

June 14, 2019

ICML 2020

Function Approximation

- **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.

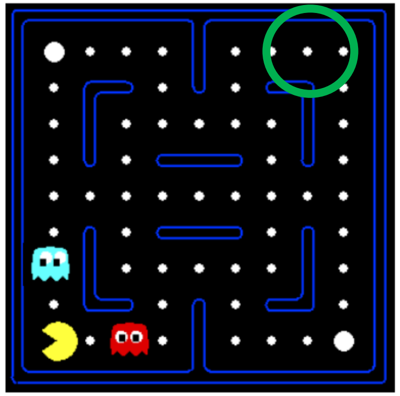


Figure 1

We discover through experience that this state is bad

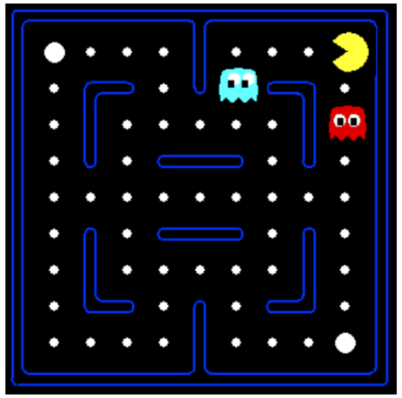


Figure 2

In tabular methods, we know nothing about this state.

Figure 3

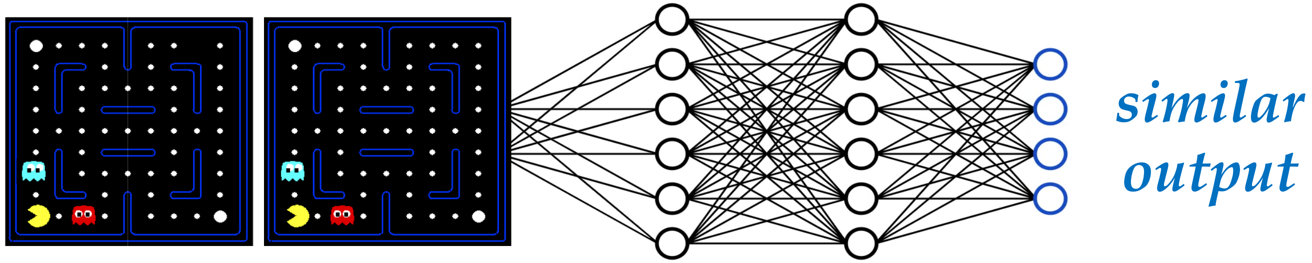
We know ***nothing*** *about this state either!*

But this way has a poor scalability in practical scenarios; and there are many structures yet to exploit…

Function Approximation

❑ **RL Function approximation:** approximate using a parameterized function.

- To avoid bad dependence on #states S, #action A in tabular MDPs
- Describe states (or state-actions) using feature representations in $\mathbb{R}^d$.
- A modern choice: DNN as a feature representer



parameterize MDP model with a low-dimensional representation

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

Deploying bandit techniques

- Linear Mixture MDPs

$$r_h(x,a) = \langle \boldsymbol{\phi}(x,a), \boldsymbol{\theta}_h^* \rangle$$

$$\mathbb{P}_h(s' \mid s,a) = \langle \psi(s' \mid s,a), \mathbf{w}_h^* \rangle$$

- $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$ is known feature map
- $\psi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ is known feature map
- $\{\boldsymbol{\theta}_h^*\}_{h=1}^H$ is the unknown reward parameter
- $\{\mathbf{w}_h^*\}_{h=1}^H$ is the unknown transition parameter

- Linear Bandits

(1) the player first chooses an arm X_t from arm set $\mathcal{X}$;

(2) and then environment reveals a reward $r_t \in \mathbb{R}$.

- Linear modeling assumption: $r_t(x) = x^\top \theta_* + \eta_t$

Linear bandits serve as a foundational tool for understanding linear mixture MDPs

Linear Mixture MDPs

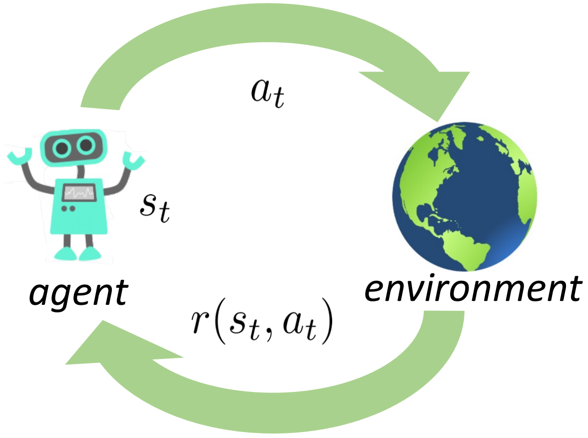
- Least square for parameter estimation

Reward estimation

$$\widehat{\boldsymbol{\theta}}_h = \arg\min_{\theta \in \mathbb{R}^d} \left\{ \frac{\lambda_\theta}{2} \|\theta\|_2^2 + \sum_{j=1}^{k-1} \left(r_h(s_h, a_h) - \phi(s_h, a_h)^\top \theta \right)^2 \right\}$$

Transition estimation

$$\widehat{\mathbf{w}}_h = \arg\min_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{\lambda_{\mathbf{w}}}{2} \|\mathbf{w}\|_2^2 + \sum_{j=1}^{k-1} \left(\langle \psi_{h+1}(s_h, a_h), \mathbf{w} \rangle - V_{h+1}(s_{h+1}) \right)^2 \right\}$$



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

Estimation error

$$\|\widehat{\mathbf{w}}_h - \mathbf{w}_h\|_{\Sigma_h} \leq \mathcal{O}\left(\sqrt{d} H (\log(t/\delta))^2 \right)$$

Regret bound

$$\text{Regret}_T \leq \widetilde{\mathcal{O}}\left(d\sqrt{H^3 K} \right)$$

K: the number of epsiodes

H: the length of each epsiode

Get back to linear bandits…

- **LinUCB** [Abbasi-Yadkori et al., NIPS 2011]

  Least-Square parameter estimation + Upper Confidence Bound Selection

Construct $\mathbf{UCB}_t(\mathbf{x})$ to ensure that $\Longrightarrow$ $X_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \mathbf{UCB}_t(\mathbf{x})$

$$\mu(\mathbf{x}) = \mathbf{x}^\top \theta_* \leq \mathbf{UCB}_t(\mathbf{x})$$

$$\triangleq \widehat{\mu}_t(\mathbf{x}) + \beta_{t-1} \|\mathbf{x}\|_{V_{t-1}^{-1}}$$

$\Uparrow$

$\Downarrow$

Submit X_{t+1},
observe $r_{t+1} \in \mathbb{R}$

$\Downarrow$

Regularized Least Square Estimator

$$\widehat{\theta}_t = \arg\min_{\theta\in\mathbb{R}^d} \lambda\|\theta\|_2^2 + \sum_{s=1}^{t-1} \left(X_s^\top \theta - r_s\right)^2$$

$$\widehat{\mu}_t(\mathbf{x}) = \mathbf{x}^\top \widehat{\theta}_t$$

$\xleftarrow{r_t = X_t^\top \theta_* + \eta_t}$

Learning History
$(X_1, r_1), \ldots, (X_t, r_t)$

Get back to linear bandits…

- **LinUCB** [Abbasi-Yadkori et al., NIPS 2011]

  Least-Square parameter estimation + Upper Confidence Bound Selection

  - *Estimator*: regularized least-square estimation

$$\widehat{\theta}_t = \arg\min_{\theta \in \mathbb{R}^d} \lambda \|\theta\|_2^2 + \sum_{s=1}^{t-1} \left(X_s^\top \theta - r_s\right)^2$$

  - *Arm selection*: upper confidence bound

$$X_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \left\{ \underbrace{\mathbf{x}^\top \widehat{\theta}_t}_{\textit{exploit}} + \underbrace{\beta_{t-1} \|\mathbf{x}\|_{V_{t-1}^{-1}}}_{\textit{explore}} \right\}$$

Optimal Regret

$$\left| \mathbf{x}^\top (\widehat{\theta}_t - \theta_*) \right| \leq \beta_{t-1} \|\mathbf{x}\|_{V_{t-1}^{-1}}$$

$$\beta_{t-1} \leq \mathcal{O}\left(\log(t-1)\right)$$

$$\mathbf{Reg}_T = \sum_{t=1}^T X_*^\top \theta_* - \sum_{t=1}^T X_t^\top \theta_* \leq \widetilde{O}(\sqrt{T})$$

LinUCB Algorithm [Abbasi-Yadkori et al., NIPS 2011]

- Regularized least-square Estimator

$$\widehat{\theta}_t = \arg\min_{\theta \in \mathbb{R}^d} \lambda \|\theta\|_2^2 + \sum_{s=1}^{t-1} \left(X_s^\top \theta - r_s\right)^2$$

Computational property:

Closed form: $\widehat{\theta}_t = V_{t-1}^{-1} b_{t-1}$

$$V_{t-1} \triangleq \lambda I + \textstyle\sum_{s=1}^{t-1} X_s X_s^\top$$

$$b_{t-1} \triangleq \textstyle\sum_{s=1}^{t-1} r_s X_s$$

"one-pass" incremental update

online data item is processed only once,
don't need to store it along the time

$$\widehat{\theta}_{t+1} = V_t^{-1} b_t, \text{ where}$$

$$V_t = V_{t-1} + X_t X_t^\top$$

$$b_t = b_{t-1} + r_t X_t^\top$$

further using rank-1 update, only $O(d^2)$ cost

$$\widehat{\theta}_{t+1} = \widehat{\theta}_t + K_{t+1}\left(r_{t+1} - X_{t+1}^\top \widehat{\theta}_t\right)$$

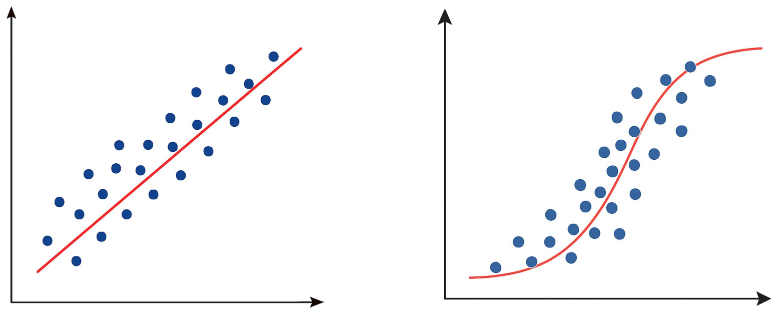
$$P_t = P_{t-1} - K_t X_t^\top P_{t-1}$$

$$K_t = \frac{P_{t-1} X_t}{1 + X_t^\top P_{t-1} X_t}$$

Beyond: More Expressivity

(i) Generalized linear bandits

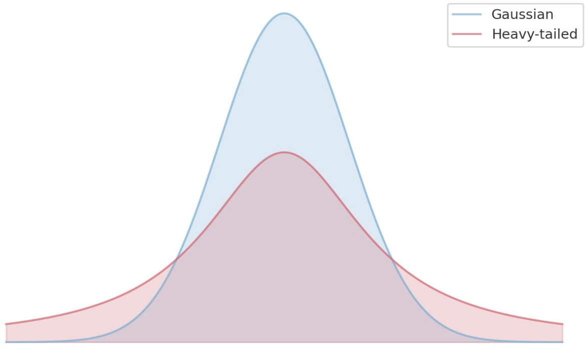
$$r_t = \mu(X_t^\top \theta_*) + \eta_t$$



Linear *Non-linear*

(ii) Heavy-tailed linear bandits

$$r_t = X_t^\top \theta_* + \eta_t$$



Goal: computationally efficient (better "one-pass") algorithm with optimal regret

- [Wang-Zhang-**Z**-Zhou, ICML'25] Heavy-Tailed Linear Bandits: Huber Regression with One-Pass Update.
- [Zhang-Xu-**Z**-Sugiyama, NeurIPS'25] Generalized Linear Bandits: Almost Optimal Regret with One-Pass Update.

① GLB: Problem Formulation

Generalized Linear Bandits

At each round $t = 1, 2, \cdots$

(1) the player first chooses an arm X_t from arm set $\mathcal{X}$;

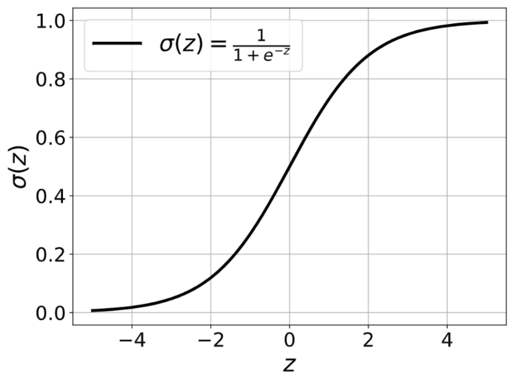
(2) and then environment reveals a reward $r_t \in \mathbb{R}$.

❑ Generalized linear reward function: $r_t = \mu(X_t^\top \theta_*) + \eta_t$

Examples: logistic bandit

$$r_t = \begin{cases} 0 \ (\text{“not click”}) & \text{w.p. } \mu(X_t^\top \theta_*) \\ 1 \ (\text{“click”}) & \text{otherwise} \end{cases}$$

$$\mu(z) = \frac{1}{1+\exp(-z)}$$



① GLB: Existing Algorithm

- GLM-UCB Algorithm [Filippi et al., NIPS 2010]
  - ***Estimator***: maximum likelihood estimator

$$\widehat{\theta}_t = \arg\min_{\theta\in\Theta} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t-1} \ell_s^{\mathrm{GLB}}(\theta), \text{ with } \ell_s^{\mathrm{GLB}}(\theta) = -\log \mathbb{P}_\theta\left(r_{s+1} \mid X_s\right)$$

<u>Estimation error:</u> $\left|\mu(\mathbf{x}^\top\widehat{\theta}_t) - \mu(\mathbf{x}^\top\theta_*)\right| \leq \frac{k_\mu}{c_\mu}\beta_{t-1}\|\mathbf{x}\|_{V_{t-1}^{-1}}$

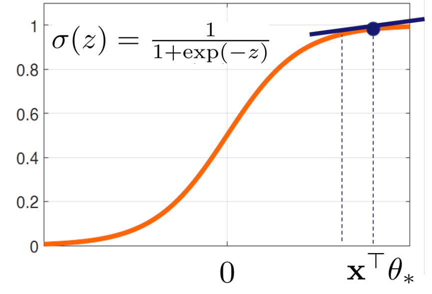
  - ***Arm selection***: upper confidence bound

$$X_t = \arg\max_{\mathbf{x}\in\mathcal{X}} \left\{\mu(\mathbf{x}^\top\widehat{\theta}_t) + \beta_{t-1}\|\mathbf{x}\|_{V_{t-1}^{-1}}\right\}$$

<u>Regret bound:</u> $\textsc{Reg}_T \leq \widetilde{\mathcal{O}}\left(\frac{k_\mu}{c_\mu} d\sqrt{T}\right)$

* Note: $c_\mu \leq \mu'(z) \leq k_\mu, \forall z \in [-S, S]$

The non-linear term k_μ/c_μ can be as large as $\mathcal{O}(e^S)$!



There are recent works using "warm-up" to remove κ, but is still not one-pass

② Hvt-LB: Problem Formulation

- Linear reward with sub-Gaussian noise $r_t = X_t^\top \theta_* + \boxed{\eta_t}$

Assumption 1 (sub-Gaussian noise). The noise η_t is conditionally R-sub-Gaussian for some $R \geq 0$ i.e.

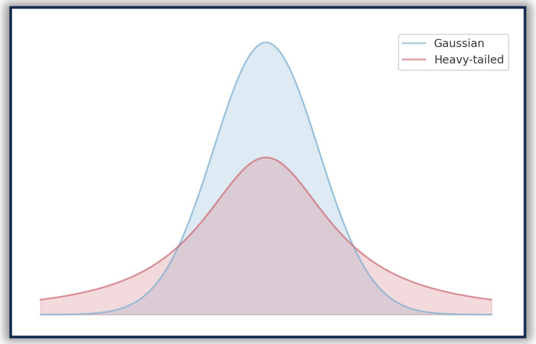
$$\forall \lambda \in \mathbb{R}, \mathbb{E}\left[\exp\left(\lambda \eta_t\right) \mid X_{1:t}, \eta_{1:t-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right).$$

In many scenarios, the noise can be heavy-tailed !

- Linear bandits with heavy-tailed noise

Assumption 2 (heavy-tailed noise). The noise $\{\eta_t, \mathcal{F}_t\}$ is is martingale difference ($\mathbb{E}\left[\eta_t \mid \mathcal{F}_{t-1}\right] = 0$), and satisfies that for some $\varepsilon \in (0, 1], \nu_t > 0$,

$$\mathbb{E}\left[\left|\eta_t\right|^{1+\varepsilon} \,\middle|\, \mathcal{F}_{t-1}\right] \leq \nu_t^{1+\varepsilon}.$$



② Hvt-LB: Existing Algorithm

- HEAVY-OFUL Algorithm [Huang et al., NeurIPS 2023]

  - ***Estimator***: adaptive Huber regression

$$\widehat{\theta}_t = \arg\min_{\theta \in \Theta} \frac{\lambda}{2} \|\theta\|_2^2 + \sum_{s=1}^{t-1} \ell_s^{\mathrm{Hvt}}(\theta)$$

<u>Estimation error:</u> $\left\| \hat{\theta}_{t+1} - \theta_* \right\|_{V_t} \leq \widetilde{\mathcal{O}}\left(t^{\frac{1-\varepsilon}{2(1+\varepsilon)}} \right)$

Huber loss is defined using a threshold $\tau_s > 0$,

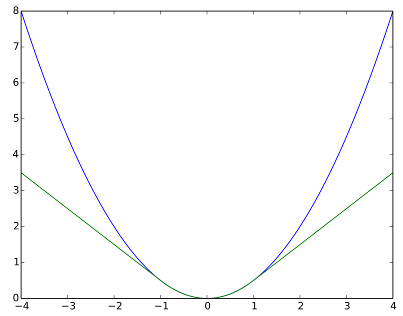
$$\ell_s^{\mathrm{Hvt}}(\theta) = \begin{cases} \frac{z_s(\theta)^2}{2} & \text{if } |z_s(\theta)| \leq \tau_s, \\ \tau_s |z_s(\theta)| - \frac{\tau_s^2}{2} & \text{if } |z_s(\theta)| > \tau_s, \end{cases}$$

with $z_s(\theta) = \frac{r_s - X_s^\top \theta}{\sigma_s}$.

  - ***Arm selection***: upper confidence bound

$$X_t = \arg\max_{\mathbf{x} \in \mathcal{X}} \left\{ \mathbf{x}^\top \widehat{\theta}_t + \beta_{t-1} \|\mathbf{x}\|_{V_{t-1}^{-1}} \right\}$$

<u>Regret bound:</u> $\textsc{Reg}_T \leq \widetilde{\mathcal{O}}\left(d T^{\frac{1}{1+\varepsilon}} \right)$



Squared loss

Huber loss

reduce penalty for large deviation

Efficiency Concerns

- **Stochastic LB**: least squares (closed-form solution)

$$\widehat{\theta}_t = \arg\min_{\theta \in \mathbb{R}^d} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t-1} \left(X_s^\top \theta - r_s\right)^2 \Longrightarrow$$

one-pass update

$$\widehat{\theta}_t = V_{t-1}^{-1}\left(\textstyle\sum_{s=1}^{t-1} r_s X_s\right)$$

$$V_{t-1} = \lambda I + \textstyle\sum_{s=1}^{t-1} X_s X_s^\top$$

- **Generalized LB**: maximum likelihood estimator

$$\widehat{\theta}_t = \arg\min_{\theta \in \Theta} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t-1} \ell_s^{\mathrm{GLB}}(\theta)$$

- **Heavy-tailed LB**: adaptive Huber regression

$$\widehat{\theta}_t = \arg\min_{\theta \in \Theta} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t} \ell_s^{\mathrm{Hvt}}(\theta)$$

$\Longrightarrow$ *inefficiency due to non-quadratic loss*

The cost at round t

Computational cost: $\mathcal{O}(t \log T)$

Storage cost: $\mathcal{O}(t)$

infeasible!

Question: Can Generalized/Heavy-tailed LB enjoy one-pass algorithms?

Outline

- Bandits Problem
- One-Pass Bandits
- Extensions
- Summary

Online Mirror Descent (OMD)

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

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

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.

We here use OMD as a statistical estimation tool!

✓ **GLB**: use OMD and exploit self-concordance property to achieve one-pass estimator with desired statistical error

✓ **Hvt-LB**: use OMD and adaptively adjust Huber loss regions to achieve one-pass estimator with desired statistical error

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	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2}\lVert \mathbf{x} - \mathbf{x}_t \rVert_2^2$	$\lVert \mathbf{x} \rVert_2^2$	$\frac{1}{\sqrt{t}}$	$\mathcal{O}(\sqrt{T})$
OGD for strongly c.	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2}\lVert \mathbf{x} - \mathbf{x}_t \rVert_2^2$	$\lVert \mathbf{x} \rVert_2^2$	$\frac{1}{\sigma t}$	$\mathcal{O}(\frac{1}{\sigma}\log T)$
ONS for exp-concave	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \frac{1}{2}\lVert \mathbf{x} - \mathbf{x}_t \rVert_{A_t}^2$	$\lVert \mathbf{x} \rVert_{A_t}^2$	$\frac{1}{\gamma}$	$\mathcal{O}(\frac{d}{\gamma}\log T)$
Hedge for PEA	$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\Delta_N} \eta_t \langle \mathbf{x}, \nabla f_t(\mathbf{x}_t) \rangle + \mathrm{KL}(\mathbf{x} \Vert \mathbf{x}_t)$	$\sum_{i=1}^N x_i \log x_i$	$\sqrt{\frac{\ln N}{T}}$	$\mathcal{O}(\sqrt{T\log N})$

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

Online Mirror Descent (OMD)

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.

- Analysis: standard regret analysis of OMD with twist yields

Lemma 1. *For OMD estimator, we have*

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

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

① Generalized Linear Bandits

- OMD-based estimator: ***curvature-aware*** local norm design

$$\theta_{t+1} = \arg\min_{\theta\in\Theta} \widetilde{\ell}_t(\theta) + \frac{1}{2\eta}\|\theta - \theta_t\|_{H_t}^2,$$

$$\widetilde{\ell}_t(\theta) \triangleq \langle \nabla \ell_t(\theta_t), \theta - \theta_t\rangle + \tfrac{1}{2}\|\theta - \theta_t\|^2_{\nabla^2 \ell_t(\theta_t)}$$

$$H_t \triangleq \lambda I_d + \textstyle\sum_{s=1}^{t-1} \nabla^2 \ell_s(\theta_{s+1})$$

Computational Efficiency

$$\zeta_{t+1} = \theta_t - \eta \widetilde{H}_t^{-1} \nabla \ell_t(\theta_t),$$

$$\theta_{t+1} = \arg\min_{\theta\in\Theta} \|\theta - \zeta_{t+1}\|^2_{\widetilde{H}_t},$$

$$\widetilde{H}_t = H_t + \eta \nabla^2 \ell_t(\theta_t)$$

***Technique**: self-concordance property, second-order approximation, lookahead regularizer, etc.*

Lemma 1 (Estimation Error). *Let the regularization parameter* $\lambda = 2\max\{7d\eta R^2, \max\{3\eta RS, 1\}C_\mu/g(\tau)\}$ *and the stepsize* $\eta = 1 + RS$. *Then, with probability at least* $1-\delta$, $\forall t > 1$, *we have with*

$$\|\theta_* - \theta_t\|_{H_t} \le \beta_t(\delta) \triangleq \sqrt{4\lambda S^2 + 2\eta \ln\left(\frac{1}{\delta}\right) + 6d\eta^2 \ln\left(2 + \frac{2C_\mu t}{\lambda g(\tau)}\right)} = \mathcal{O}\left(SR\sqrt{d\left(S^2 R + \ln\frac{t}{\delta}\right)}\right).$$

① Generalized Linear Bandits

GLM-UCB

MLE $\widehat{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t} \ell_s(\theta)$

Comp. cost per round $\mathcal{O}(t)$

Estimation error $\mathcal{O}\left(\kappa\sqrt{d\log t}\right)$

GLB-OMD

OMD $\widehat{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \widetilde{\ell}_t(\theta) + \frac{1}{2\eta}\|\theta - \widehat{\theta}_t\|_{H_t}^2$

Comp. cost per round $\mathcal{O}(1)$

Estimation error $\mathcal{O}\left(\sqrt{d\log t}\right)$

one-pass!

Theorem 2. *With probability at least* $1-\delta$*, the regret of* GLB-OMD *with parameter* $\eta = 1 + RS$ *and* $\lambda = 2\max\{7d\eta R^2, \max\{3\eta RS, 1\}C_\mu/g(\tau)\}$ *ensures*

$$\textsc{Reg}_T \lesssim dSR\sqrt{S^2R+\log T}\sqrt{\frac{T\log T}{\kappa_*}} + \kappa d^2S^2R^3\log T(S^2R+\log T),$$

① Generalized Linear Bandits

- Our work improves upon previous works **with a novel mixability-based analysis**
  - *Statistical efficiency*: maintain the optimal and instant-dependent regret bound
  - *Computational efficiency*: reduce the per round time and storage cost

Method	Regret	Time per Round	Memory	Jointly Efficient
GLM-UCB [Filippi et al., 2010]	$\mathcal{O}(\kappa(\log T)^{\frac{3}{2}}\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$	✗
GLOC [Jun et al., 2017]	$\mathcal{O}(\kappa \log T\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✗
OFUGLB [Lee et al., 2024, Liu et al., 2024]	$\mathcal{O}(\log T\sqrt{T/\kappa_*})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$	✗
RS-GLinCB [Sawarni et al., 2024]	$\mathcal{O}(\log T\sqrt{T/\kappa_*})$	$\mathcal{O}\big((\log t)^2\big)^\dagger$	$\mathcal{O}(t)$	✗
GLB-OMD (Theorem 2 of this paper)	$\mathcal{O}(\log T\sqrt{T/\kappa_*})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	✓

The first one-pass GLB algorithm with (almost) optimal regret guarantee!

[Zhang-Xu-**Z**-Sugiyama, NeurIPS'25] Generalized Linear Bandits: Almost Optimal Regret with One-Pass Update.

② Heavy-Tailed Bandits

- OMD-based estimator: ***curvature-aware*** local norm design

$$\widehat{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \left\{ \left\langle \theta, \nabla \ell_t(\widehat{\theta}_t) \right\rangle + \mathcal{D}_{\psi_t}(\theta, \widehat{\theta}_t) \right\}$$

$$\psi_t(\theta) = \frac{1}{2}\|\theta\|_{V_t}^2 \text{ with } V_t \triangleq \lambda I + \frac{1}{\alpha}\sum_{s=1}^{t} \frac{X_s X_s^\top}{\sigma_s^2}$$

Computational Efficiency

$$\widetilde{\theta}_{t+1} = \widehat{\theta}_t - V_t^{-1}\nabla \ell_t(\widehat{\theta}_t)$$

$$\widehat{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \left\| \theta - \widetilde{\theta}_{t+1} \right\|_{V_t}$$

***Technique**: adaptively adjust the threshold/renormalized factor in Huber loss, exploit curvature of in/out-liers*

Lemma 1 (Estimation error). *If σ_t, τ_t, τ_0 are set as where $w_t \triangleq \frac{1}{\sqrt{\alpha}} \left\| \frac{X_t}{\sigma_t} \right\|_{V_{t-1}^{-1}}$ and let the step size $\alpha = 4$, then with probability at least $1 - 4\delta, \forall t \geq 1$, we have*

$$\|\widehat{\theta}_{t+1} - \theta_*\|_{V_t} \leq \beta_t \triangleq 107 \log \frac{2T^2}{\delta} \tau_0 t^{\frac{1-\varepsilon}{2(1+\varepsilon)}} + \sqrt{\lambda\left(2 + 4S^2\right)}$$

② Heavy-Tailed Bandits

HEAVY-OFUL

MLE $\widehat{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t} \ell_s(\theta)$

Comp. cost per round $\mathcal{O}(t)$

Estimation error $\widetilde{\mathcal{O}}\left(t^{\frac{1-\epsilon}{2(1+\epsilon)}}\right)$

Hvt-UCB

OMD $\widehat{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \left\{\langle\theta, \nabla\ell_t(\widehat{\theta}_t)\rangle + \frac{1}{2}\|\theta - \widehat{\theta}_t\|_{V_t}^2\right\}$

Comp. cost per round $\mathcal{O}(1)$

Estimation error $\widetilde{\mathcal{O}}\left(t^{\frac{1-\epsilon}{2(1+\epsilon)}}\right)$ one-pass!

Theorem 4. *By setting $\sigma_t, \tau_t, \tau_0, \alpha$ as in Lemma 1, and let $\lambda = d, \sigma_{\min} = \frac{1}{\sqrt{T}}, \delta = \frac{1}{8T}$, with probability at least $1 - 1/T$, the regret of* Hvt-UCB *is bounded by*

$$\textsc{Reg}_T \leq \widetilde{\mathcal{O}}\left(dT^{\frac{1-\varepsilon}{2(1+\varepsilon)}}\sqrt{\sum_{t=1}^{T}\nu_t^2} + dT^{\frac{1-\varepsilon}{2(1+\varepsilon)}}\right).$$

When $\nu_t = \nu$, this can recover to optimal regret bound $\textsc{Reg}_T \leq \widetilde{\mathcal{O}}\left(dT^{\frac{1}{1+\varepsilon}}\right)$

② Heavy-Tailed Bandits

- Our work maintains the regret with only $O(1)$ computational cost.

Method	Algorithm	Regret	Comp. cost	Remark
MOM	MENU [Shao et al., 2018] CRMM [Xue et al., 2023]	$\tilde{\mathcal{O}}\left(dT^{\frac{1}{1+\varepsilon}}\right)$	$\mathcal{O}(\log T)$ $\mathcal{O}(1)$	fixed arm set and repeated pulling
Truncation	TOFU [Shao et al., 2018] CRTM [Xue et al., 2023]	$\tilde{\mathcal{O}}\left(dT^{\frac{1}{1+\varepsilon}}\right)$	$\mathcal{O}(t)$ $\mathcal{O}(1)$	absolute moment $\mathbb{E}[\lvert r_t\rvert^{1+\varepsilon} \mid \mathcal{F}_{t-1}] \leq u$
Huber	HEAVY-OFUL [Huang et al., 2024]	$\tilde{\mathcal{O}}\left(dT^{\frac{1-\varepsilon}{2(1+\varepsilon)}}\sqrt{\sum_{t=1}^{T}\nu_t^2} + dT^{\frac{1-\varepsilon}{2(1+\varepsilon)}}\right)$	$\mathcal{O}(t\log T)$	instance-dependent bound
Huber	Hvt-UCB (Corollary 1)	$\tilde{\mathcal{O}}\left(dT^{\frac{1}{1+\varepsilon}}\right)$	$\mathcal{O}(1)$	$\mathbb{E}[\lvert\eta_t\rvert^{1+\varepsilon} \mid \mathcal{F}_{t-1}] \leq \nu^{1+\varepsilon}$
Huber	Hvt-UCB (Theorem 1)	$\tilde{\mathcal{O}}\left(dT^{\frac{1-\varepsilon}{2(1+\varepsilon)}}\sqrt{\sum_{t=1}^{T}\nu_t^2} + dT^{\frac{1-\varepsilon}{2(1+\varepsilon)}}\right)$	$\mathcal{O}(1)$	instance-dependent bound

The first one-pass algorithm for heavy-tailed linear bandits with (almost) optimal regret!

[Wang-Zhang-**Z**-Zhou, ICML'25] Heavy-Tailed Linear Bandits: Huber Regression with One-Pass Update.

Outline

- Bandits Problem
- One-Pass Bandits
- RL Implications
- Summary

Implication 1. 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

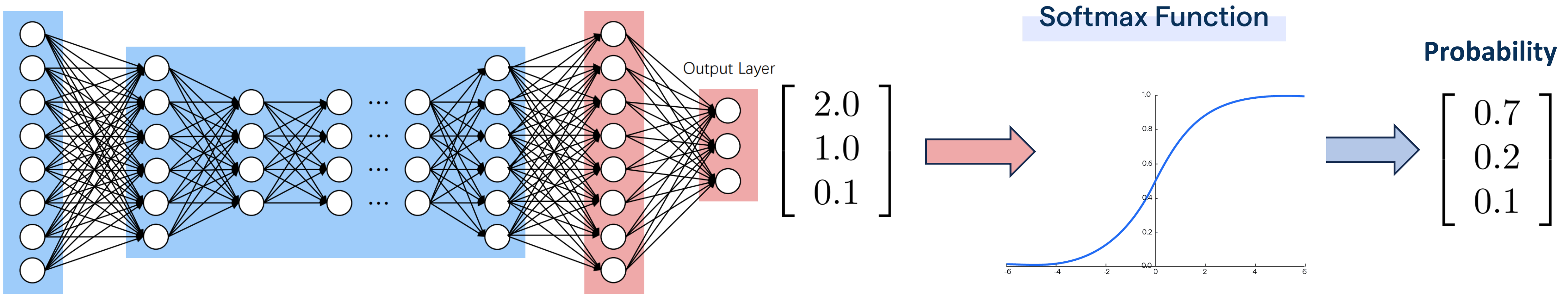
- Eluder dimension [Russo and Roy, 2013, Jin et al., 2021]
- Decision-Estimation Coefficient (DEC) [Foster et al., 2021]
- Admissible Bellman Characterization (ABC) [Chen et al., 2023]
- …

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

Deploying bandits techniques

- **Multinomial Logistic (MNL) Mixture MDP**

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

- $\phi(s'|s,a)$ 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

- **Multinomial Logistic Bandit** (a special case of generalized linear bandits)

$$r_t = \begin{cases} 0 & (\text{“feedback } y_t = 0\text{”}) \\ \rho_1 & \boxed{(\text{“feedback } y_t = 1\text{”})} \\ \ldots & \\ \rho_K & (\text{“feedback } y_t = K\text{”}) \end{cases}$$

The feedback y_t from environments is generated by the multinomial logit model:

$$\Pr[y_t = k \mid \mathbf{x}_t] = \frac{\exp(\mathbf{x}_t^\top \mathbf{w}_k^*)}{1+\sum_{j=1}^K \exp(\mathbf{x}_t^\top \mathbf{w}_j^*)}$$

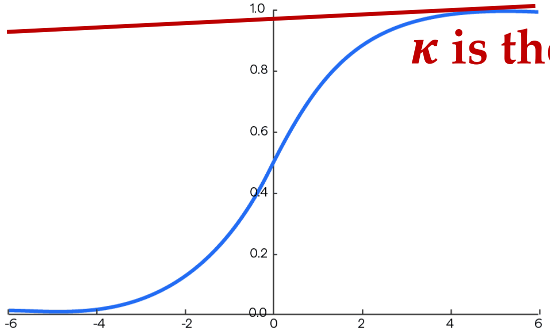
where $\mathbf{w}_k^* \in \mathbb{R}^d$ is the parameter for $y_t = k$

Key Challenge: non-linearity

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

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

Softmax Function



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

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

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

MNL Mixture MDPs

- OMD for one-pass estimation

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

one-pass!

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.

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	$\widetilde{\mathcal{O}}(dH^2\sqrt{K} + \kappa^{-1}d^2H^2)$	$\Omega(dH\sqrt{K})$

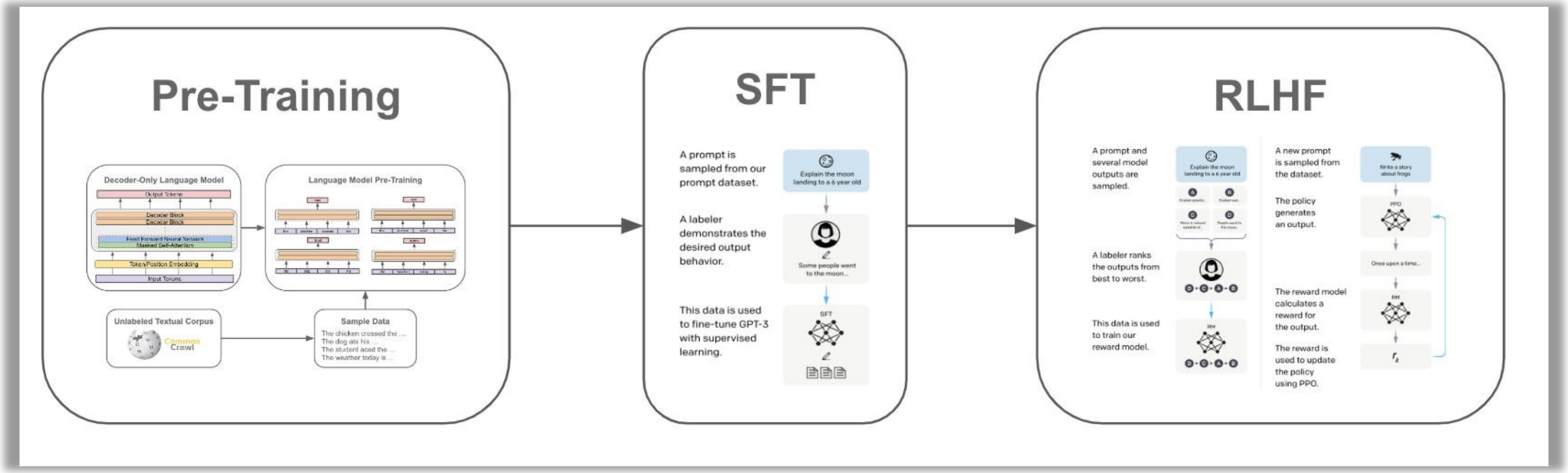
in the worst case, $\kappa^{-1} = \Omega(S^2)$

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

[Li-Zhang-**Z**-Zhou, NeurIPS'24] Provably Efficient Reinforcement Learning with Multinomial Logit Function Approximation.

Implication 2. RLHF

❑ 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 Formulation

- **Input:** a 4-argument preference 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

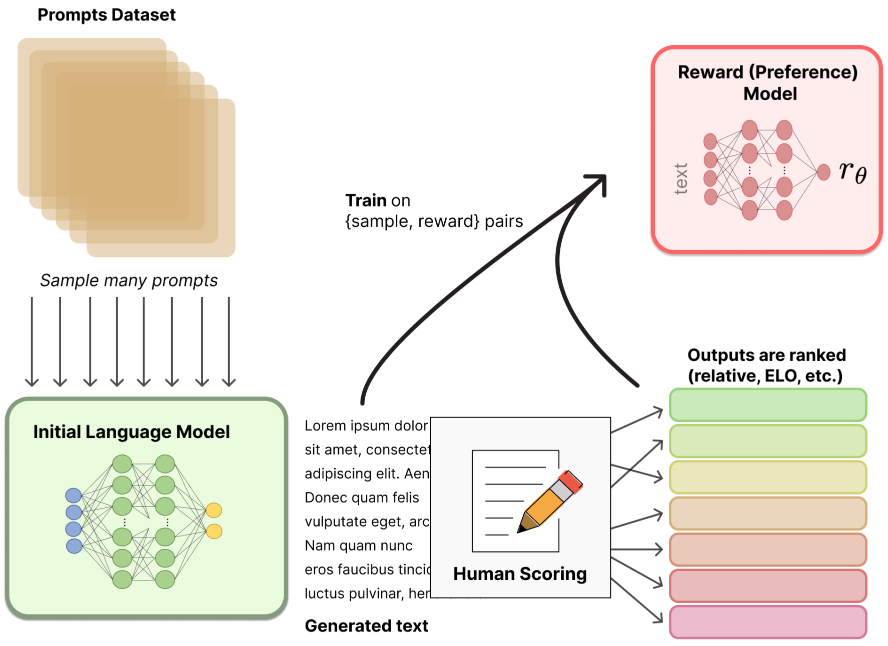
  i.e., ***align LLM with human's preference or value (encoded in the preference data)***

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

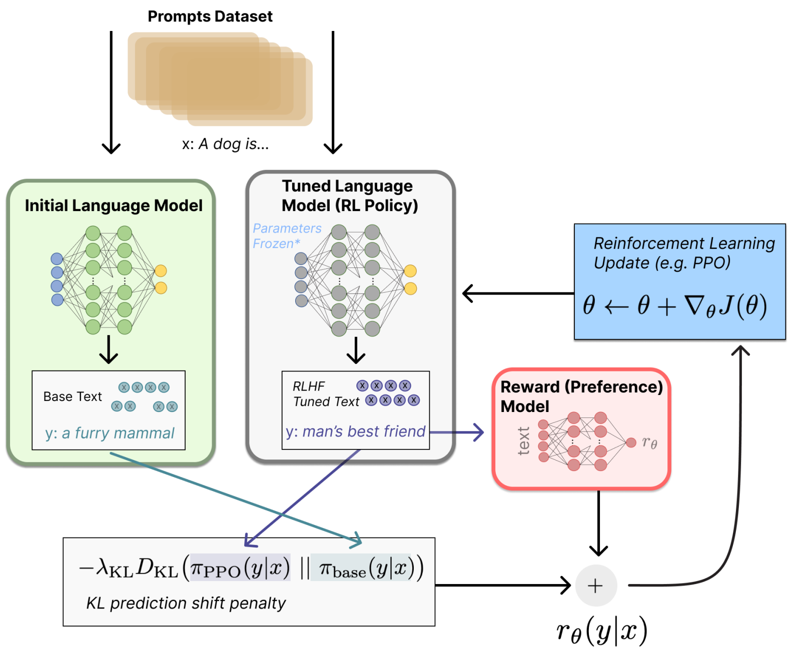
RLHF for Alignment

- A standard pipeline of RLHF: reward modelling + PPO

(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

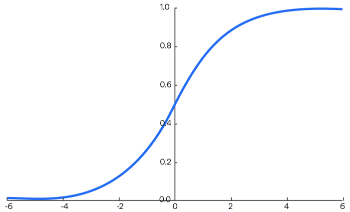
$$\mathbb{P}\left(a \succ a' \mid x\right) = \frac{\exp\left(r\left(x, a\right)\right)}{\exp\left(r\left(x, a\right)\right) + \exp\left(r\left(x, a'\right)\right)}$$

where r is the latent function.

- **Maximum Likelihood Estimation (MLE)**

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

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



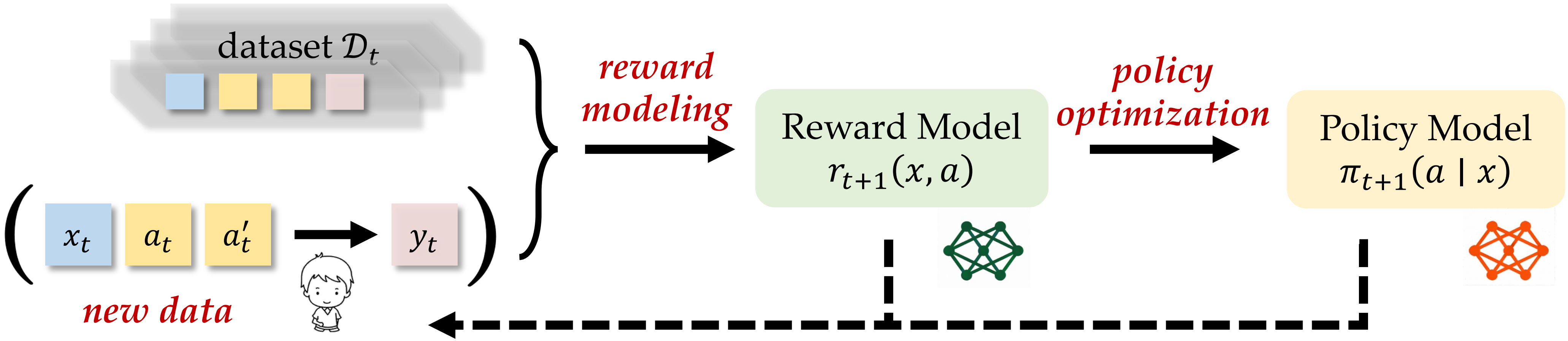
Online RLHF

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: $O(t)$

Deploying bandits techniques

- **Linear reward model assumption**

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

BT model $\Longrightarrow$

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

- $\phi(x,a)$ is the known feature mapping
- θ^* is the unknown parameter

- **Contextual dueling bandits**

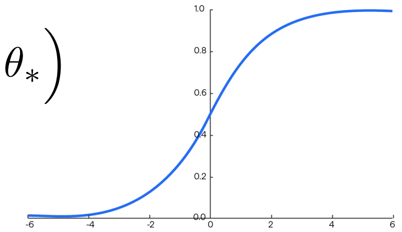
At each round $t = 1, 2, \cdots$

(1) the learner first chooses two arms $\mathbf{x}_t, \mathbf{y}_t \in \mathcal{X} \subseteq \mathbb{R}^d$;

(2) and then environment reveals a preference feedback o_t.

$$\mathbb{P}\left(o_t = 1\right) = \mu\left(\left(\mathbf{x}_t - \mathbf{y}_t\right)^\top \theta_*\right)$$

$$\mu(z) = \frac{1}{1 + \exp(-z)}$$



One-Pass Reward Modeling

- OMD for one-pass estimation

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

$$\widetilde{\theta}_{t+1} = \underset{\theta \in \Theta}{\arg\min} \left\{ \langle g_t(\widetilde{\theta}_t), \theta \rangle + \frac{1}{2\eta} \|\theta - \widetilde{\theta}_t\|_{\widetilde{\mathcal{H}}_t}^2 \right\}, \text{ where } \widetilde{\mathcal{H}}_t = \underbrace{\sum_{i=1}^{t-1} H_i(\widetilde{\theta}_{i+1})}_{\text{look-ahead local norm}} + \underbrace{\eta H_t(\widetilde{\theta}_t)}_{\text{second-order approximation}} + \lambda I.$$

Constant time and storage complexity,
Independent of t

one-pass!

look-ahead local norm

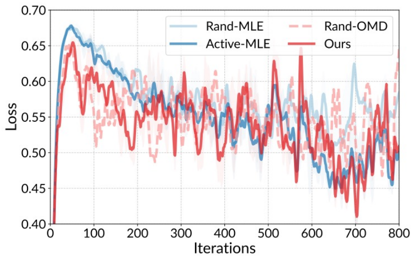
second-order approximation

Estimation error

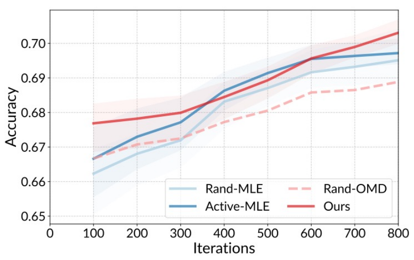
$$\|\theta - \widetilde{\theta}_t\|_{\mathcal{H}_t} \leq \mathcal{O}\left(\sqrt{d}(\log(t/\delta))^2\right)$$

Regret bound

$$\mathrm{Reg}_T \leq \widetilde{\mathcal{O}}\left(d\sqrt{\frac{T}{\kappa}}\right)$$



(a) training loss



(b) evaluation accuracy

[Li*-Qian*-Z-Zhou, NeurIPS'25] Provably Efficient Online RLHF with One-Pass Reward Modeling.

Outline

- Bandits Problem

- One-Pass Bandits

- RL Implications

- Summary

Summary

❑ One-Pass Bandits

- Beyond linear bandits: For non-quadratic loss, MLE doesn't enjoy the one-pass property
- ***Generalized linear bandits***: exploit the self-concordance property of the link function
- ***Heavy-tailed linear bandits***: adaptively set Huber threshold to adjust curvatures such that outliers fall in the linear region, while normal data remain in the quadratic region

❑ OMD Estimator

- Online Mirror Descent as a statistical estimator, where the ***curvature-aware adaptivity*** is crucial for the local norm design; similar to "from SGD to AdaGrad/Adam"

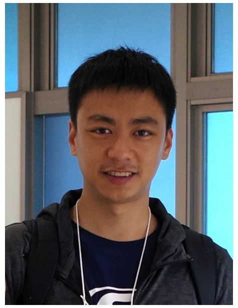
❑ RL Implications

- ***RL with function approximation***: MNL mixture MDPs (related to GLB)
- ***RLHF***: BT model naturally related to logistic bandits, etc.

One-Pass Bandits: Reference

- Yu-Jie Zhang, Sheng-An Xu, Peng Zhao, Masashi Sugiyama. Generalized Linear Bandits: Almost Optimal Regret with One-Pass Update. NeurIPS 2025.
- Long-Fei Li*, Yu-Yang Qian*, Peng Zhao, Zhi-Hua Zhou. Provably Efficient Online RLHF with One-Pass Reward Modeling. NeurIPS 2025.
- Jing Wang, Yu-Jie Zhang, Peng Zhao, and Zhi-Hua Zhou. Heavy-Tailed Linear Bandits: Huber Regression with One-Pass Update. ICML 2025.
- Long-Fei Li, Yu-Jie Zhang, Peng Zhao, Zhi-Hua Zhou. Provably Efficient Reinforcement Learning with Multinomial Logit Function Approximation. NeurIPS 2024.

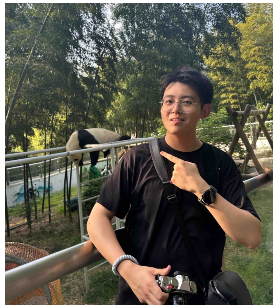
Thanks!



Yu-Jie Zhang
(NJU → U Tokyo → UW)

Jing Wang
(NJU)

Long-Fei Li
(NJU→ Noah's Ark Lab)



Yu-Yang Qian
(NJU)

Sheng-An Xu
(NJU → UCB)

Zhi-Hua Zhou
(NJU)