One-Pass Bandit Learning

for RLHF and Function Approximation

Peng Zhao

School of AI, Nanjing University

June 18, 2026

Outline

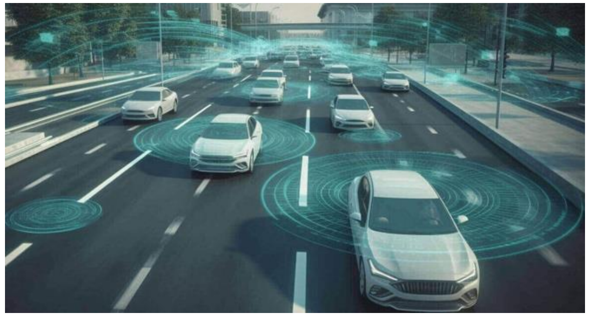
- Interactive Learning
- One-Pass Bandits
- RL Implications
- Summary

Interactive Machine Learning

- Many ML systems are now ***interactive*** with environment/user…

recommendation

agentic system



self-driving car

- Interactive Learning: ***effective*** and ***efficient*** over **long horizon**
  - online horizon T can be very large (millions/billions of rounds), or even unbounded
  - impossible to store all historical interactions (memory cost, privacy, etc).

One-Pass Learning

- **"Re-fitting" fails at scale**: a standard estimator at round t

regularized ERM/MLE: $$\widehat{\theta}_t \in \arg\min_{\theta \in \Theta} \sum_{s=1}^{t-1} \ell_s(\theta) + \lambda \Omega(\theta).$$

- **One-Pass Learning**

Definition (One-Pass Estimator) An estimator $\widehat{\theta}_t$ is called ***one-pass*** if there exists a state S_t such that:

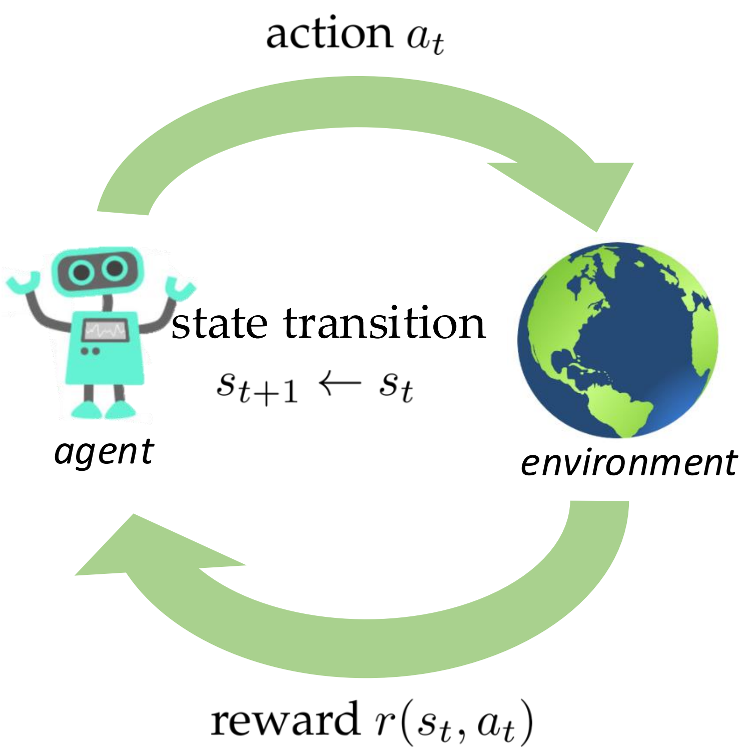
- S_t has a fixed size of poly(d);
- $S_{t+1} = \text{OnlineUpdate}\left(S_t, \mathbf{z}_t\right)$, and then *discard* $\mathbf{z}_t$;
- $\widehat{\theta}_t = \mathcal{G}\left(S_t\right)$, and both update and computation are *independent of* t.

✓ per-round time/storage memory: poly(d), independent of t

✓ no historical data kept (esp. privacy issue..)

Bandits: Interactive Learning

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



Reinforcement learning:

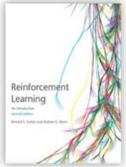
- Sequential decision making
- With state transition

Bandits:

- Single-step decision making
- No state transition

Contents

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



Bandits

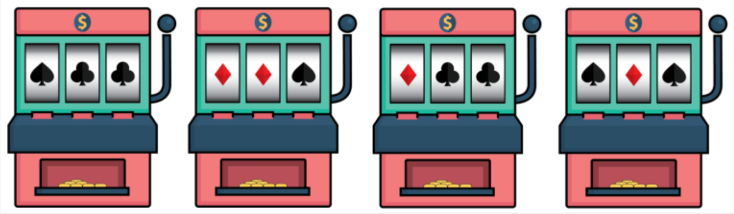
❑ Multi-armed bandits: a simplest formulation for bandit problems

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 $(a_t, r_t(a_t))$.



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

❑ Linear Bandits: context matters (especially important for ML)

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

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

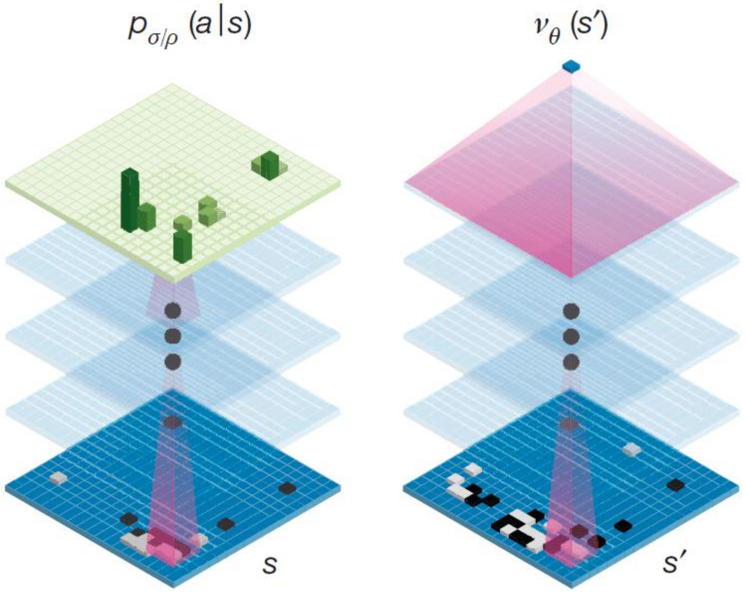
– reward paramterized by an unknown parameter θ_*;

– with random noise: η_t is sub-Gaussian noise

- Example: book recommendation
- Feature: Each arm is a book w. side information

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

Bandits: Estimate-Action

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

  Paramter estimation + Upper Confidence Bound Selection; $\widetilde{\mathcal{O}}(d\sqrt{T})$ near-optimal regret

Construct $\mathbf{UCB}_t(\mathbf{x})$ as
$$\mathbf{x}^\top \widehat{\theta}_t + \beta_{t-1}\|\mathbf{x}\|_{V_{t-1}^{-1}}$$
$$\geq \mathbf{x}^\top \theta_* \text{ (expected reward)}$$
$$V_t = \lambda I + \textstyle\sum_{s\leq t} X_s X_s^\top$$

⇒

Select the arm by
$$X_{t+1} = \operatorname*{arg\,max}_{\mathbf{x}\in\mathcal{X}} \mathbf{UCB}_t(\mathbf{x})$$
Submit X_{t+1},
observe $r_{t+1} \in \mathbb{R}$

⇓

Learning History
$$(X_1, r_1), \ldots, (X_t, r_t)$$
$$r_t = X_t^\top \theta_* + \eta_t$$

⇐

Parameter Estimation
Estimate $\widehat{\theta}_t$ using history data till t

⇑

➢**Action**: upper confidence bound
$$X_{t+1} = \operatorname*{arg\,max}_{\mathbf{x}\in\mathcal{X}} \Big\{ \underbrace{\mathbf{x}^\top \widehat{\theta}_t}_{\textit{exploit}} + \underbrace{\beta_{t-1}\|\mathbf{x}\|_{V_{t-1}^{-1}}}_{\textit{explore}} \Big\}$$

estimation error relates to exploration

➢**Estimator**: regularized least-square
$$\widehat{\theta}_t = \operatorname*{arg\,min}_{\theta\in\mathbb{R}^d} \lambda\|\theta\|_2^2 + \sum_{s=1}^{t-1}\left(X_s^\top \theta - r_s\right)^2$$

LinUCB is Naturally One-Pass

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

a special gift of quadratic loss

GIFT

sufficient statistics

$$V_t = \lambda I + \textstyle\sum_{s \le t} X_s X_s^\top$$
$$b_t = \textstyle\sum_{s \le t} r_s X_s^\top$$

✓ **Online Update**

$$\widehat{\theta}_{t+1} = V_t^{-1} b_t$$
$$V_t = V_{t-1} + X_t X_t^\top$$
$$b_t = b_{t-1} + r_t X_t^\top$$

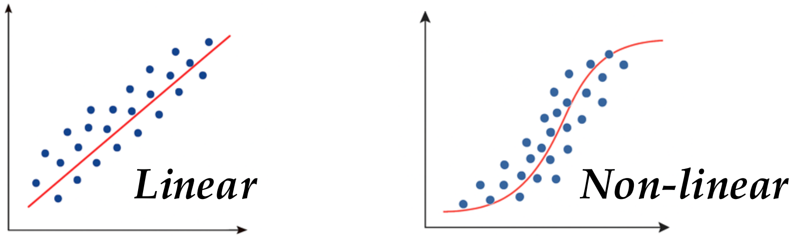
✓ **Rank-1 Update: only $O(d^2)$ cost**

$$\widehat{\theta}_{t+1} = \widehat{\theta}_t + K_{t+1}\left(r_{t+1} - X_{t+1}^\top \widehat{\theta}_t\right)$$
$$P_t = P_{t-1} - K_t X_t^\top P_{t-1},$$
$$K_t = P_{t-1} X_t \cdot (1 + X_t^\top P_{t-1} X_t)^{-1}$$

- Beyond linear bandits: More Expressivity

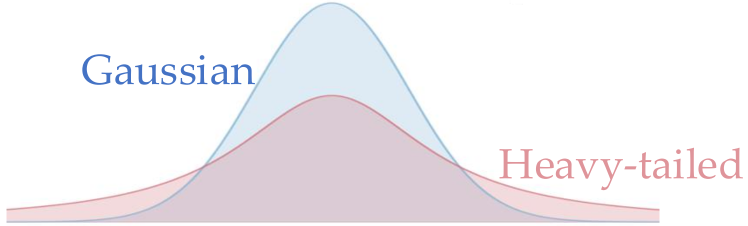
(i) Generalized linear bandits

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



(ii) Heavy-tailed linear bandits

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



① Generalized Linear Bandits

- Generalized linear reward modeling:

$$\mathbb{E}[r \mid x;\theta] = \mu\left(x^\top\theta\right)$$

  - Linear: $\mu(z) = z$
  - Logistic: $\mu(z) = 1/(1+\exp(-z))$
  - Poisson: $\mu(z) = \exp(z)$

Logistic model: binary feedback (click/not-click, prefer/not-prefer)

$$r_t \in \{0,1\}, \quad \mathbb{P}\left[r_t = 1 \mid x_t\right] = \mu\left(x_t^\top\theta_*\right).$$

logistic (Bradley–Terry) modeling $\mu(z) = 1/(1+\exp(-z))$

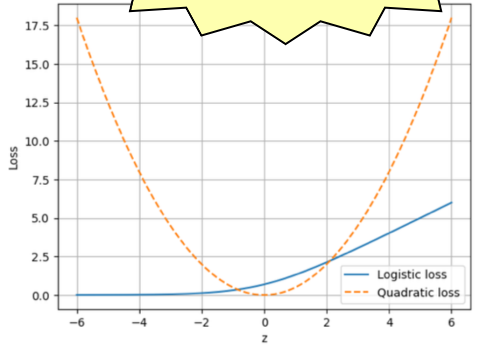
- GLM-UCB [Filippi et al., NIPS 2010]
  - ➢ **Estimator**: regularized maximum likelihood estimator

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

$$\ell_s^{\mathrm{logistic}}(\theta) = \log\left(1 + e^{x_s^\top\theta}\right) - r_{s+1}x_s^\top\theta$$

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

convex but not quadratic!



② Heavy-Tailed Bandits

- Heavy-tailed reward modeling:

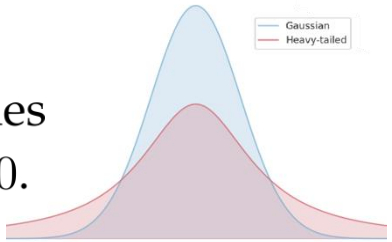
$$r_t = X_t^\top \theta_* + \eta_t$$

with heavy-tailed noise $\{\eta_t\}$

Esp. data contamination/large noise (e.g., in finance, sensor/IoT data).

Heavy-tailed noise, with $(1+\varepsilon)$-moment bounded

The noise $\{\eta_t\}$ is is martingale difference, and satisfies that $\mathbb{E}[|\eta_t|^{1+\varepsilon} \mid \mathcal{F}_{t-1}] \le \nu_t^{1+\varepsilon}$ for some $\varepsilon \in (0,1]$, $\nu_t > 0$.



- Heavy-tail: only a low-order moment is bounded (even variance unstable)
- Sub-Gaussian: exponential tail decay → all moments are controlled

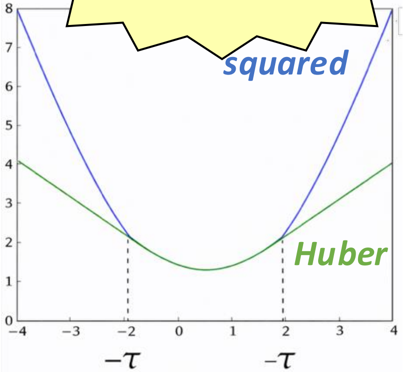
- HEAVY-OFUL [Huang et al., NeurIPS 2023]

  - **Estimator**: adaptive Huber regression (reduce penalty for large deviation)

$$\widehat{\theta}_t = \arg\min_{\theta\in\Theta} \frac{\lambda}{2}\|\theta\|_2^2 + \sum_{s=1}^{t-1} \ell_s^{\mathrm{Hvt}}(\theta), \text{ with } \ell_s^{\mathrm{Hvt}}(\theta) = \begin{cases} \frac{z_s(\theta)^2}{2} & \text{if } |z_s(\theta)| \le \tau_s, \\ \tau_s|z_s(\theta)| - \frac{\tau_s^2}{2} & \text{if } |z_s(\theta)| > \tau_s, \end{cases}$$

Huber loss is defined using a threshold $\tau_s > 0$, with $z_s(\theta) = \frac{r_s - X_s^\top \theta}{\sigma_s}$.

only partially quadratic!



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

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

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

- **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

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

convex but non-quadratic loss

$\Longrightarrow$ **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?

SGD for One-Pass Update?

Can we use SGD to incrementally solve MLE?

e.g., $\widehat{\theta}_{t+1} = \widehat{\theta}_t - \eta_t \nabla \ell_t(\widehat{\theta}_t)$

➢**Action**: upper confidence bound

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

estimation error relates to exploration

➢**Estimator**: regularized least-square

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

Why plain SGD is not enough?

1) optimization error typically in the form of $\|\widehat{\theta}_t - \theta_*\|_2^2$

2) i.i.d. over samples $\{(x_s, r_s)\}_{s\le t}$

bandits need estimation for:

1) action selection needs a ***"self-normalized"*** V_t-exploration:

$$\left|\mathbf{x}^\top(\widehat{\theta}_t - \theta_*)\right| \le \beta_t \|\mathbf{x}\|_{V_t^{-1}} \quad \text{where } V_t = \lambda I + \sum_{s\le t} X_s X_s^\top$$

2) ***on-policy*** issue in bandits/RL

$$X_t = \pi_t\left(\widehat{\theta}_1, \ldots, \widehat{\theta}_{t-1}\right), \quad \widehat{\theta}_t = \text{Update}(\widehat{\theta}_{t-1}, X_t)$$

no longer i.i.d., so classical guarantees are not applied

Our Method: Online Mirror Descent

- OMD is a powerful framework for online regret optimization.

consider both ***gradient update*** and ***geometry***

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

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.

	$\psi(\mathbf{x})$	$\mathcal{D}_\psi(\mathbf{x}, \mathbf{y})$
Squared L_2-distance	$\|\mathbf{x}\|_2^2$	$\|\mathbf{x} - \mathbf{y}\|_2^2$
Mahalanobis distance	$\|\mathbf{x}\|_A^2$	$\|\mathbf{x} - \mathbf{y}\|_A^2$
Negative entropy	$\sum_i x_i \log x_i$	$\mathrm{KL}(\mathbf{x}\|\mathbf{y})$

Online Mirror Descent

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

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

Advanced Optimization (Fall 2025) Lecture 7. Online Mirror Descent 24

More details of OMD can be found in Lecture 7 of Advanced Optimization Course 2025 Fall
https://www.pengzhao-ml.com/course/AOpt2025fall/

Curvature-aware OMD Estimator

OMD with local norm (curvature-aware geometry matrix)

- local norm: typically choose $\psi_t(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_{H_t}^2$

$$\theta_{t+1} = \Pi_{\Theta}^{H_t}\left[\theta_t - \eta H_t^{-1} g_t\right], \quad g_t = \nabla \ell_t(\theta_t)$$

- Exploitation: using g_t
- Exploration: encoding H_t

- curvature geometry: using a weighted rank-1 increment to keep as a state

$$H_{t+1} = H_t + \alpha_t X_t X_t^\top$$

require a careful design

- complexity: independent of t

rank-1 updates $\Rightarrow$ time per round $O(d^2)$, memory $O(d^2)$

✓ **GLB**: use OMD and exploit self-concordance property to design the geometry matrix.

✓ **Hvt-LB**: use OMD and adaptively adjust Huber loss regions to set the weight.

"Self-Normalized" Error Analysis

- Standard regret analysis of OMD with twist yields

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

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

Lemma 1. *For OMD estimator, we have*

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

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

***Importantly**: compatible with self-normalized concentration*
key to address on-policy dependence (details omitted…)

① Generalized Linear Bandits

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

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

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

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

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*

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

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*

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

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

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

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

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*

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

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

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.

RL 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

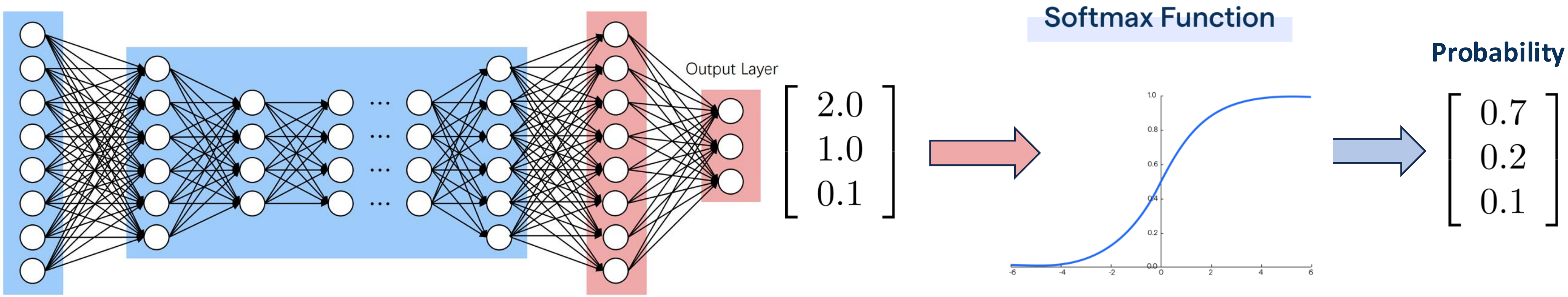
- Eluder dimension [Russo and Roy, 2013, Jin et al., 2021]
- Decision-Estimation Coefficient (DEC) [Foster et al., 2021]
- Admissible Bellman Characterization (ABC) [Chen et al., 2023]
- …

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:

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

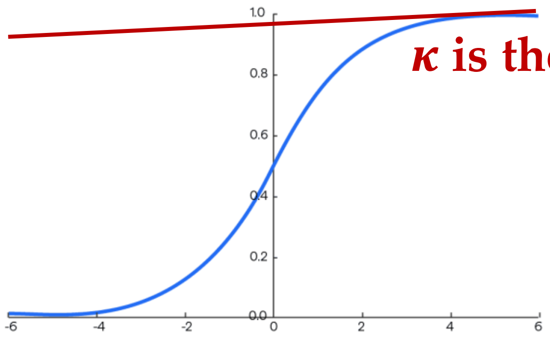
- $\phi(s'|s,a)$ is the known feature mapping
- $\{\theta_h^*\}_{h=1}^H$ is the unknown transition parameter
- $\mathcal{S}_{h,s,a} \triangleq \{s' \in \mathcal{S} \mid \mathbb{P}_h(s'|s,a) \neq 0\}$ is reachable states

Key Challenge: non-linearity

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

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

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

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

MNL Mixture MDPs

- OMD for one-pass estimation

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

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.

RL Implication 2. RLHF

RLHF (or *preference optimization*): align model towards human preferences or values.

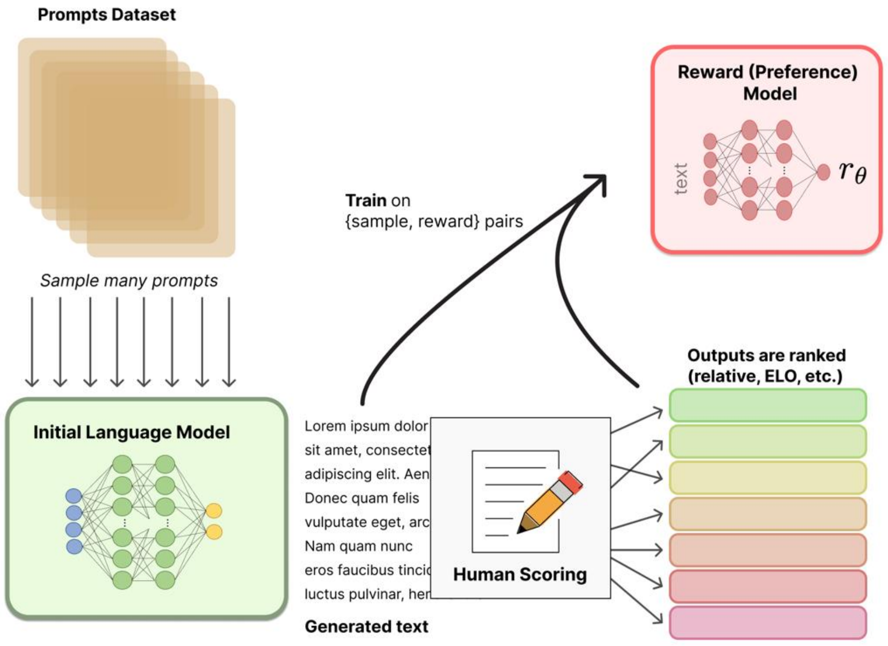
- **Input:** a 4-argument preference tuple $(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{a}', \boldsymbol{y})$
  - $\boldsymbol{x}$: the prompt: `"Please write a joke for me."`
  - $\boldsymbol{a}$: the first response: `"Sorry, I can't."`
  - $\boldsymbol{a}'$: the second response: `"Here is a joke for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

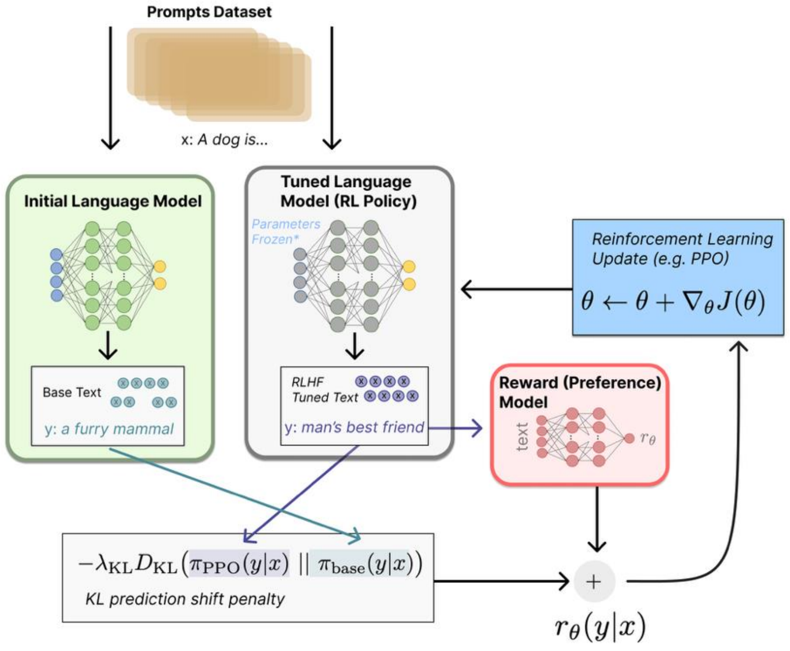
RLHF for Alignment

- A standard pipeline of RLHF: reward modelling + PPO

(i) reward model learning



(ii) policy optimization (guided by reward model)



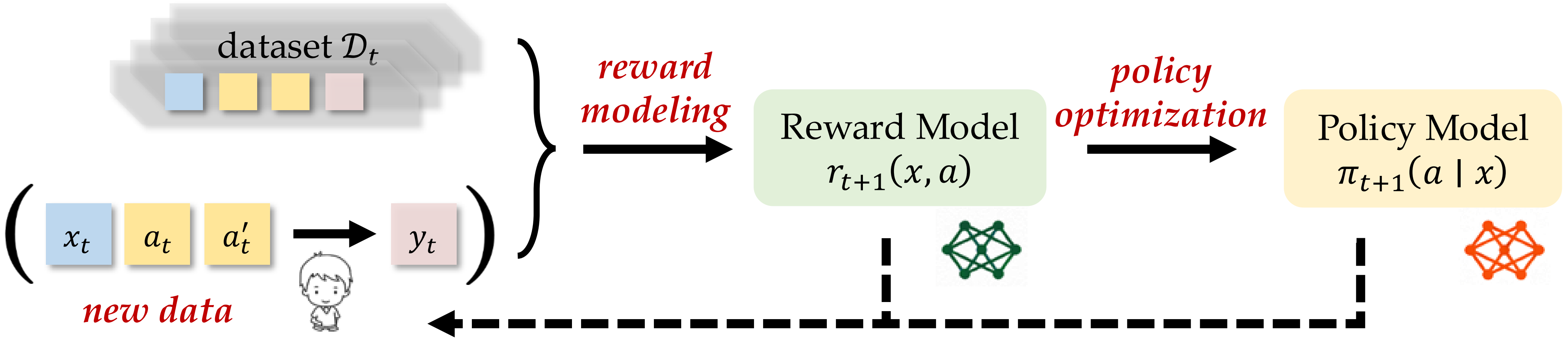
Online RLHF

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 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

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

where r is the latent function.

- **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:

per-round time: $O(t \log t)$,

storage memory: $O(t)$

Deploying bandits techniques

- **Linear reward model assumption**

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

BT model

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

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

- **Contextual dueling bandits**

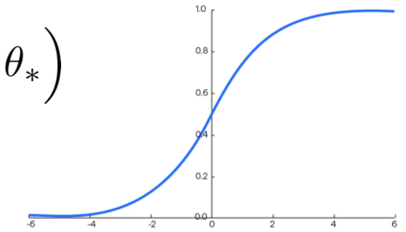
At each round $t = 1, 2, \cdots$

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

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

constant time and storage cost, independent of t

one-pass!

look-ahead local norm

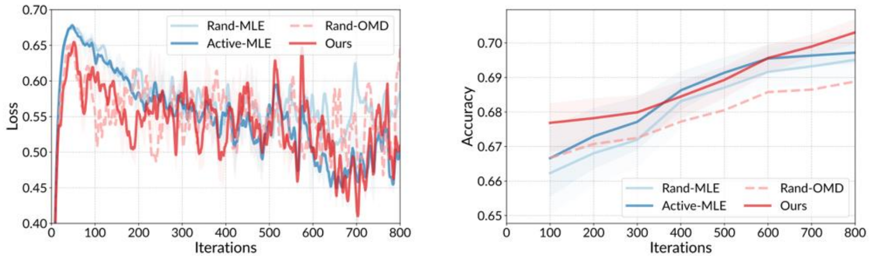
second-order approximation

Estimation error

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

Regret bound

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



(a) training loss

(b) evaluation accuracy

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

Summary

How to do interactive learning without revisiting history with guarantee?

One-Pass Bandits and RL

- 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
- **RL implications:** *RL with function approximation*: MNL mixture MDPs (related to GLB); *RLHF*: Bradley-Terry model naturally relates to logistic bandits, etc.

Technical takeaway: 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"

Reference

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

Thanks!

Some follow-up works

- S Kim, J Lee, and SY Yun. A Jointly Efficient and Optimal Algorithm for Heteroskedastic Generalized Linear Bandits with Adversarial Corruptions. Arxiv Preprint: 2602.10971.

The aforementioned discussions lead to the following central question:

*Can we design a computationally **efficient** algorithm for (self-concordant) **heteroskedastic GLBs with adversarial corruptions** that **generically** attains instance-wise minimax-optimal regret bounds?*

We answer this question affirmatively, almost. Our contributions are as follows:

1. We propose `HCW-GLB-OMD`, a Hessian-based confidence-weighted (HCW) version of the recently proposed `GLB-OMD` (Zhang et al., 2025b). Despite its simplicity, we show that leading-term-wise, it attains the following regret bound:

$$\mathrm{REG}(T) \lesssim_{\log} d\sqrt{\sum_{t=1}^{T} g(\tau_t)\dot{\mu}_{t,\star}} + (d^2 g_{\max} + dC)\kappa,$$

for *any* self-concordant ***heteroskedastic GLBs with adversarial corruptions*** (**Theorem 6**).

Some follow-up works

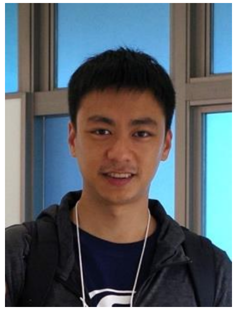
- Joongkyu Lee, and Min-hwan Oh. Nonstationary Generalized Linear Bandits with Discounted Online Mirror Descent. Arxiv Preprint: 2605.25590.

Can we design a nonstationary GLB algorithm with constant computation and memory costs?

In this paper, we answer this question affirmatively by proposing `DOMD-GLB`, a discounted OMD (DOMD)-based algorithm that achieves both statistical and computational efficiency.

Technical difficulties. Establishing this result is nevertheless nontrivial, since the stationary OMD analysis of Zhang et al. [2025] does not directly extend to the discounted setting. The first challenge lies in the *mirror-descent geometry*. In OMD, the choice of Bregman divergence is crucial, since it determines the geometry with respect to which the update is performed. For stationary GLBs [Zhang et al., 2025], the additively accumulated Hessian naturally induces the relevant mirror-descent geometry. In the nonstationary setting, however, this geometry must account for forgetting: past curvature should be downweighted over time, rather than accumulated uniformly. Consequently, the stationary geometry cannot be directly applied to the discounted update structure. As a result, it is not clear *a priori* what the correct discounted analogue of the stationary Bregman divergence should be. Establishing such a mirror-descent geometry is therefore a central difficulty in extending OMD to nonstationary GLBs.

Joint works with



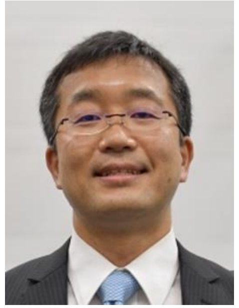
Yu-Jie Zhang
(NJU → U Tokyo → UW)

Jing Wang
(NJU)

Long-Fei Li
(NJU→ Noah's Ark Lab)

Yu-Yang Qian
(NJU)

Sheng-An Xu
(NJU → UCB)



Masashi Sugiyama
(RIKEN & U Tokyo)

Zhi-Hua Zhou
(NJU)

Thanks!