Generalized Linear Bandits: Almost Optimal Regret with One-Pass Update

Yu-Jie Zhang, Sheng-An Xu, Peng Zhao, Masashi Sugiyama

Advanced OPT, Dec 26 2025

Outline

- Generalized Linear Bandits
- Statistical and Computational Efficient Challenge
- Jointly efficient Method
- Conclusion

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*** :

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

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

Stochastic Linear Bandits

- A ubiquitous problem in real life: ***feature information***

  - Each arm represent a book and has side information;
  - Arm set could be very large or even infinite.

Stochastic LB: Formulation

Stochastic Linear Bandits

Each arm is associated with a feature vector $\mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq L\}$

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 $r_t \in \mathbb{R}$.

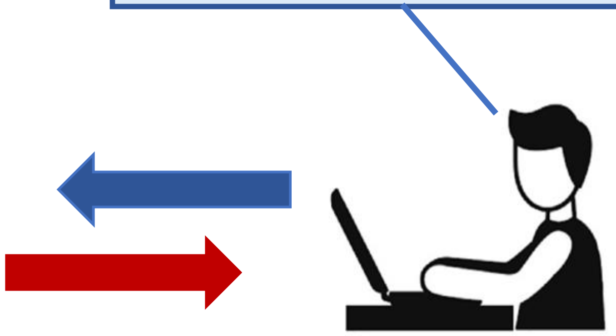
- **Linear modeling assumption:** $r_t = \mathbf{x}_t^\top \mathbf{w}_\star + \varepsilon_t$
  - for some unknown parameter $\mathbf{w} \in \mathcal{W} = \mathbf{w} \mid \|\mathbf{w}\|_2 \leq S$
  - for some unknown noise: ε_t is R-sub-Gaussian random noise;

Going Beyond Linear Bandits?

We need more expressive models beyond linear classes

The feedback is discrete:

$$\text{reward: } r_t = \begin{cases} 1 \ (\text{“buy”}) \\ 0 \ (\text{“not buy”}) \end{cases}$$



selected arm $\mathbf{x}_t$

customer with preference $\boldsymbol{\theta}_*$

Generalized Linear Bandits

Generalized linear bandits: natural exponential-family (NEF) rewards

$$\mathbb{P}(r_t | z_t = \mathbf{x}_t^\top \mathbf{w}_*) = e^{r_t z_t - m(z_t) + h(r_t)}$$

$h(r)$: base measure shaping the distribution

$m(z)$: log-partition function for normalization

- **Linear Bandit**: real value feedback $r_t = \mathbf{x}_t^\top \mathbf{w}_\star + \varepsilon_t$
- **Logistic Bandit**: binary feedback with the logit model

$$r_t = \begin{cases} 1 \text{ ("click")} & \text{w.p. } \mu(\mathbf{x}_t^\top \mathbf{w}_\star) \\ 0 \text{ ("not click")} & \text{otherwise} \end{cases}$$

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

Generalized Linear Bandits

Generalized linear bandits: natural exponential-family (NEF) rewards

$$\mathbb{P}(r_t | z_t = \mathbf{x}_t^\top \mathbf{w}_*) = e^{r_t z_t - m(z_t) + h(r_t)}$$

$h(r)$: base measure shaping the distribution

$m(z)$: log-partition function for normalization

- **Linear Bandit**: real value feedback $r_t = \mathbf{x}_t^\top \mathbf{w}_\star + \varepsilon_t$
- **Logistic Bandit**: binary feedback with the logit model
- **Poisson Bandits:** count-based feedback with unbounded reward!

$r_t \in \{0, 1, 2, \ldots\}$ drawn from: $r_t \sim \text{Poisson}(\mu(x_t^\top \mathbf{w}_*))$ $\mu(z) = \exp(z)$

Generalized Linear Bandits

Generalized linear bandits: natural exponential-family (NEF) rewards

$$\mathbb{P}(r_t | z_t = \mathbf{x}_t^\top \mathbf{w}_*) = e^{r_t z_t - m(z_t) + h(r_t)}$$

$\boldsymbol{h(r)}$: base measure shaping the distribution

$\boldsymbol{m(z)}$: log-partition function for normalization

NEF properties

Mean: $\mathbb{E}[r_t \,|\, \mathbf{x}_t^\top \mathbf{w}_*] = m'(\mathbf{x}_t^\top \mathbf{w}_*) = \mu(\mathbf{x}_t^\top \mathbf{w}_*)$

Variance: $\mathrm{Var}[r_t \,|\, \mathbf{x}_t^\top \mathbf{w}_*] = m''(\mathbf{x}_t^\top \mathbf{w}_*) = \mu'(\mathbf{x}_t^\top \mathbf{w}_*)$

$$r_t = \mu(\mathbf{x}_t^\top \mathbf{w}_*) + \varepsilon_t$$

another formulation

Generalized Linear Bandits

- Goal: select the action $\mathbf{x}_t$ that achieves the maximum expected reward.

$$\mathbb{E}\left[\sum_{t=1}^{T} r_t \middle| \mathbf{x}_t\right] = \sum_{t=1}^{T} \mu(\mathbf{x}_t^\top \mathbf{w}_*)$$

$\mu(z) = 1/(1+\exp(-z))$ is the probability of $r_t = 1$

Generalized Linear Bandits

- Goal: select the action $\mathbf{x}_t$ that achieves the maximum expected reward.

$$\mathbb{E}\left[\sum_{t=1}^{T} r_t \middle| \mathbf{x}_t\right] = \sum_{t=1}^{T} \mu(\mathbf{x}_t^\top \mathbf{w}_*)$$

$\mu(z) = 1/(1+\exp(-z))$ is the probability of $r_t = 1$

- Equal to **minimize the regret:**

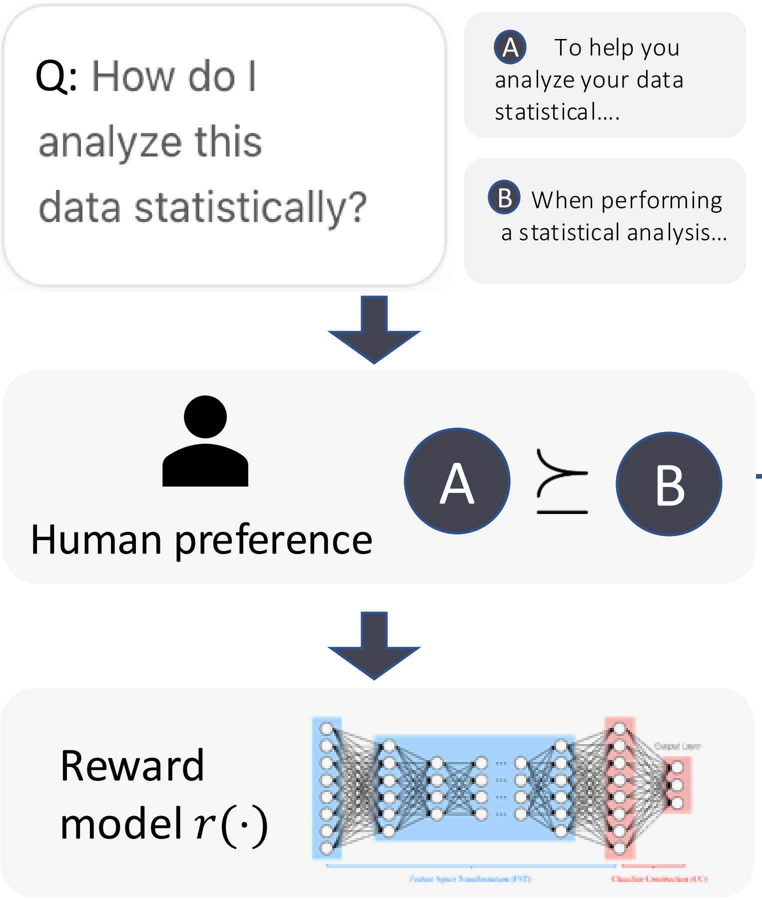
$$\text{Regret} = T \max_{\mathbf{x}\in\mathcal{X}} \boldsymbol{\sigma}(\mathbf{x}^\top \mathbf{w}_*) - \sum_{t=1}^{T} \sigma(\mathbf{x}_t^\top \mathbf{w}_*)$$

reward of the best action

reward of our algorithm

Why GLB?

Learn from human preference in dueling bandits and RLHF: **Bradley-Terry Model**



Bradley-Terry Model

$$\Pr[\mathbf{x}_1 \succcurlyeq \mathbf{x}_2] = \frac{\exp\left(r(\mathbf{x}_1)\right)}{\exp\left(r(\mathbf{x}_1)\right) + \exp\left(r(\mathbf{x}_2)\right)}$$

The issue of κ appears in many work on dueling bandits and RLHF: [Saha NeurIPS’21; Zhu et al., ICML’23; Das et al., ICML’24 workshop; Pásztor et al., NeurIPS’24; Scheid et al., arXiv’24]

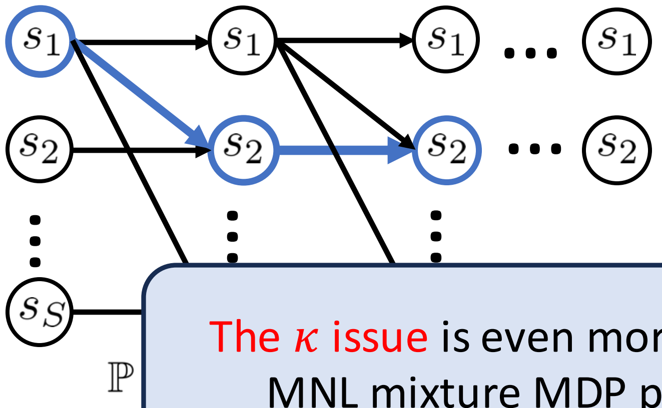
Why GLB?

To deal with large-scale MDPs: **Function Approximation**

Markov Decision Process

For iterations from $t = 1, \ldots, T$

- **Learner:** observes states $s_t \in \mathcal{S}$ and plays action $a_t \in \mathcal{A}$
- **Environments:** generates next state $s_{t+1} \sim \mathbb{P}(\cdot \mid s_t, a_t)$



The κ issue is even more severe in MNL mixture MDP problem!

MNL mixture MDPs to ensure valid distribution:

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

[Hwang and Oh et al., 2022; Li-Z-Zhao-Zhou, 2024]

Outline

- Logistic Bandits Problem
- Statistical and Computational Efficient Challenge
- Our jointly efficient Method
- Extension to Logistic Function Approximation

GLB: Existing Algorithm

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

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

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

GLB: Existing Algorithm

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

$$\widehat{\mathbf{w}}_t = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{s=1}^{t-1} \ell_s^{\mathrm{GLB}}(\mathbf{w}), \text{ with } \ell_s^{\mathrm{GLB}}(\mathbf{w}) = -\log \mathbb{P}_{\mathbf{w}}(r_{s+1} \mid \mathbf{x}_s)$$

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

degree of nonlinearity

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

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

Regret bound: $\mathrm{REG}_T \leq \widetilde{\mathcal{O}}\left(\frac{k_\mu}{c_\mu} d\sqrt{T}\right)$

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

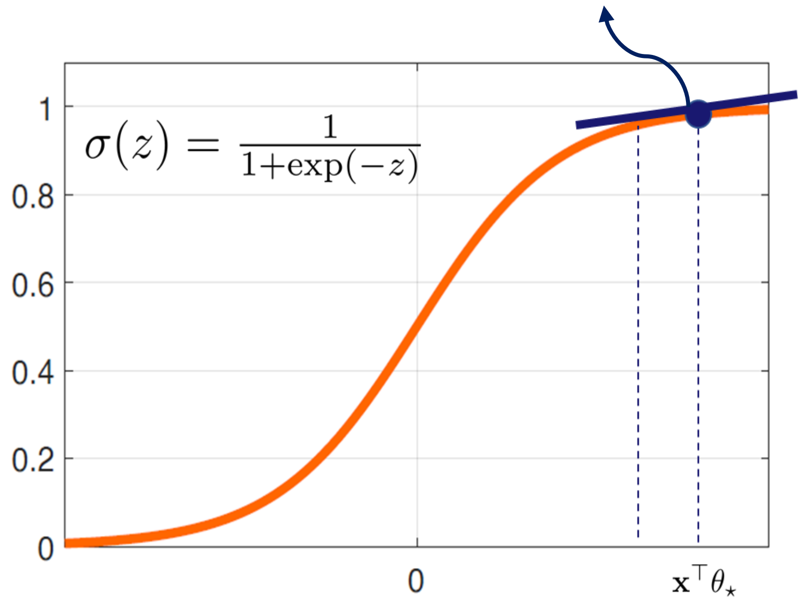
Statistical Challenge

The condition number k_μ / c_μ could be exponentially large!

Example: binary logistic bandit

- Reward function: $r(\mathbf{x}) = \sigma(\mathbf{x}^\top \mathbf{w}_*)$

$\kappa = 1/c_\mu$ is 1 divided the **minimum slope**



similar issue for Poisson bandits!

Statistical Challenge

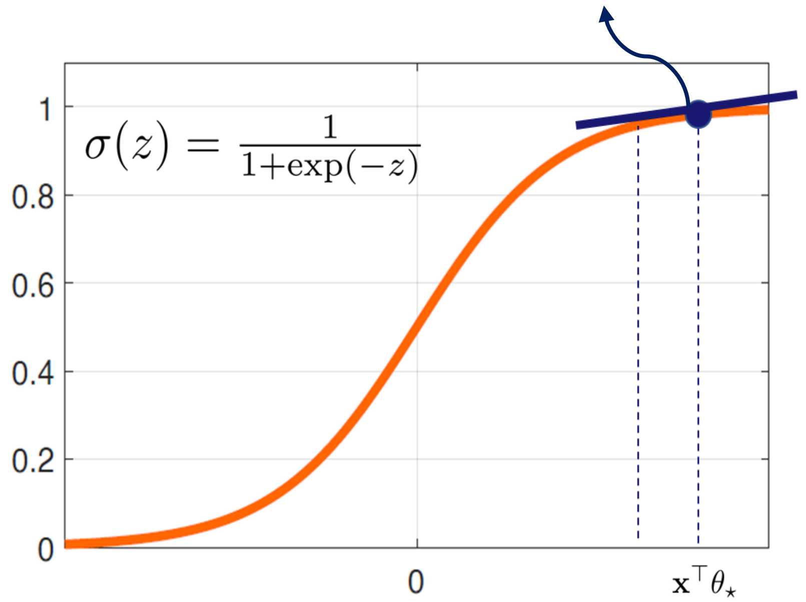
The condition number k_μ / c_μ could be exponentially large!

Example: binary logistic bandit

- Reward function: $r(\mathbf{x}) = \sigma(\mathbf{x}^\top \mathbf{w}_*)$
- GLM-UCB [Filippi et al., 2010] ensures:

$$\mathrm{REG}_T \le \widetilde{\mathcal{O}}\left(\frac{k_\mu}{c_\mu} d\sqrt{T}\right)$$

$\kappa = 1/c_\mu$ is 1 divided the **minimum slope**



similar issue for Poisson bandits!

Statistical Challenge

The condition number k_μ / c_μ could be exponentially large!

Example: binary logistic bandit

- Reward function: $r(\mathbf{x}) = \sigma(\mathbf{x}^\top \mathbf{w}_*)$
- GLM-UCB [Filippi et al., 2010] ensures:

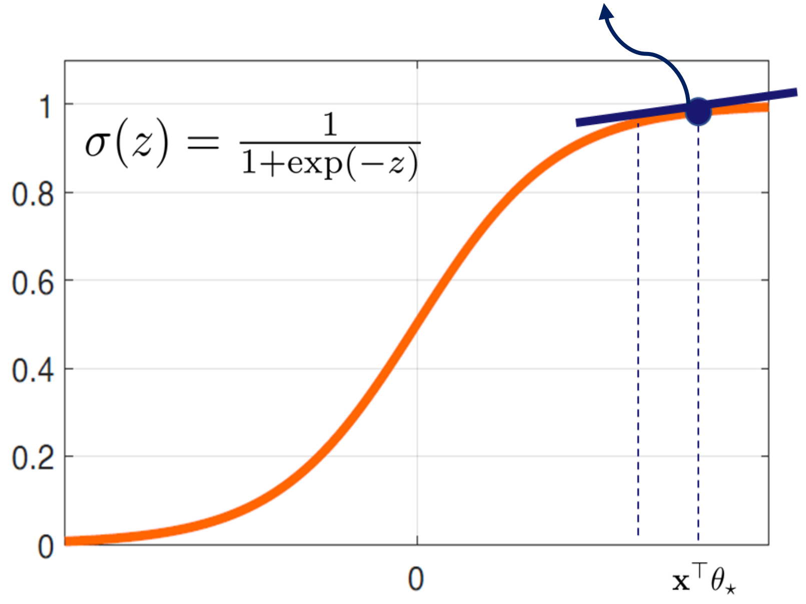
$$\text{REG}_T \leq \widetilde{\mathcal{O}}\left(\frac{k_\mu}{c_\mu} d\sqrt{T}\right)$$

- In the above, the constant

$$\kappa = \max_{\mathbf{x}\in\mathcal{X}} 1/\dot{\sigma}(\mathbf{x}^\top \mathbf{w}_*) = \mathcal{O}\left(e^{\|\mathbf{w}_*\|_2}\right)$$

is exponentially large w.r.t. $\|\mathbf{w}_*\|_2$

$\kappa = 1/c_\mu$ is 1 divided the **minimum slope**



similar issue for Poisson bandits!

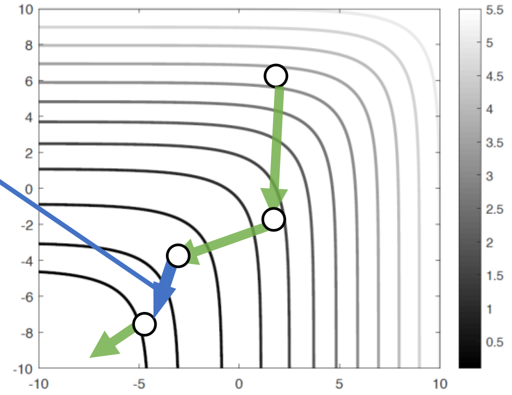
Computational Challenge

- *Maximum likelihood estimation is computationally inefficient*

$$\mathbf{w}_t^{\mathrm{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \text{ , where } \ell_t(\mathbf{w}) = -r_t\mathbf{x}_t^\top \mathbf{w} + m_t(\mathbf{w})$$

Per gradient descent step:

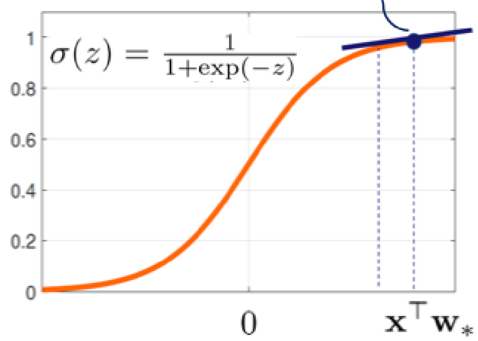
- $\mathcal{O}(t)$ time complexity per step
- $\mathcal{O}(t)$ storage complexity per step



Statistical and Computational Efficiency Concern

Setting	Algorithm	Regret	Comput. per Round	Storage Cost
linear	OFUL [Abbasi-Yadkori et al., 2011]	$\widetilde{\mathcal{O}}(\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
generalized linear	GLM-UCB [Filippi et al., 2010]	$\widetilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$

Nonlinearity of the reward function raises concerns about both statistical and computational efficiency!

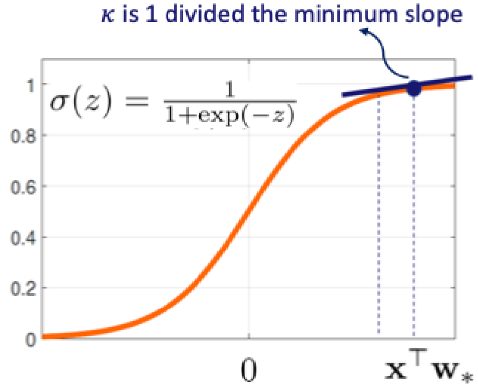


- $\kappa = \max_{\mathbf{x}\in\mathcal{X}} 1/\dot{\sigma}(\mathbf{x}^\top \mathbf{w}_*)$

Statistical and Computational Efficiency Concern

Setting	Algorithm	Regret	Comput. per Round	Storage Cost
linear	OFUL [Abbasi-Yadkori et al., 2011]	$\widetilde{\mathcal{O}}(\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
generalized linear	GLM-UCB [Filippi et al., 2010]	$\widetilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	GLOC [Jun et al., 2017]	$\widetilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

statistically inefficient computationally efficient



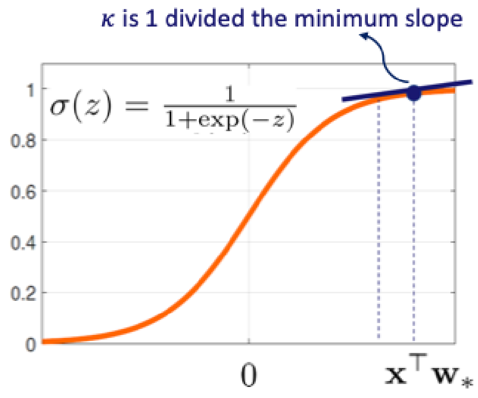
- $\kappa = \max_{\mathbf{x}\in\mathcal{X}} 1/\dot{\sigma}(\mathbf{x}^\top \mathbf{w}_*)$

Statistical and Computational Efficiency Concern

Setting	Algorithm	Regret	Comput. per Round	Storage Cost
linear	OFUL [Abbasi-Yadkori et al., 2011]	$\tilde{\mathcal{O}}(\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
generalized linear (GLB)	GLM-UCB [Filippi et al., 2010]	$\tilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	GLOC [Jun et al., 2017]	$\tilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
	OFUGLB [Lee et al., 2024; Liu et al., 2024]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	RS-GLinCB [Sawarni et al., 2024]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}((\log t)^2)^\dagger$	$\mathcal{O}(t)$

nearly minimax optimal

computationally inefficient

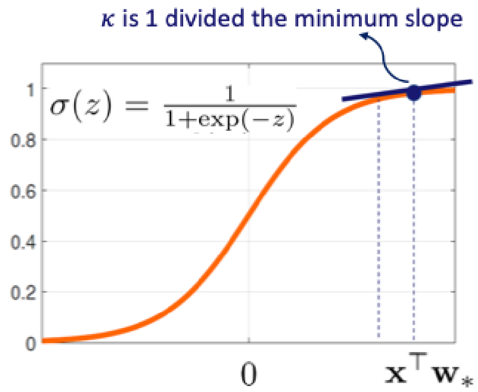


- $\kappa = \max_{\mathbf{x}\in\mathcal{X}} 1/\dot{\sigma}(\mathbf{x}^\top \mathbf{w}_*)$
- $\kappa_* = 1/\dot{\sigma}(\mathbf{x}_*^\top \mathbf{w}_*)$ is 1 over the slope at the optimal arm $\mathbf{x}_* = \arg\max_{\mathbf{x}\in\mathcal{X}} \sigma(\mathbf{x}^\top \mathbf{w}_*)$.

Statistical and Computational Efficiency Concern

Setting	Algorithm	Regret	Comput. per Round	Storage Cost
linear	OFUL [Abbasi-Yadkori et al., 2011]	$\tilde{\mathcal{O}}(\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
generalized linear (GLB)	GLM-UCB [Filippi et al., 2010]	$\tilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	GLOC [Jun et al., 2017]	$\tilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
	OFUGLB [Lee et al., 2024; Liu et al., 2024]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	RS-GLinCB [Sawarni et al., 2024]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}((\log t)^2)^\dagger$	$\mathcal{O}(t)$
	GLB-OMD [**Z**-Xu-Zhao-Sugiyama, 2025]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

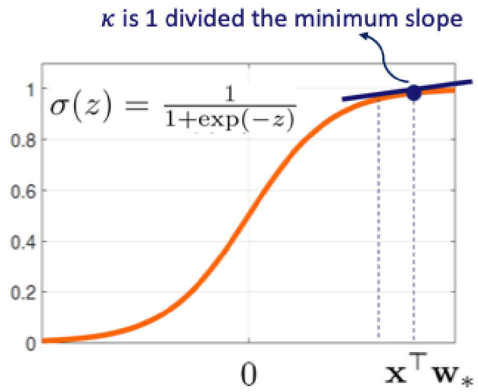
Our jointly efficient alg.!



- $\kappa = \max_{\mathbf{x}\in\mathcal{X}} 1/\dot{\sigma}(\mathbf{x}^\top \mathbf{w}_*)$
- $\kappa_* = 1/\dot{\sigma}(\mathbf{x}_*^\top \mathbf{w}_*)$ is 1 over the slope at the optimal arm $\mathbf{x}_* = \arg\max_{\mathbf{x}\in\mathcal{X}} \sigma(\mathbf{x}^\top \mathbf{w}_*)$.

Statistical and Computational Efficiency Concern

Setting	Algorithm	Regret	Comput. per Round	Storage Cost
linear	OFUL [Abbasi-Yadkori et al., 2011]	$\tilde{\mathcal{O}}(\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
generalized linear (GLB)	GLM-UCB [Filippi et al., 2010]	$\tilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	GLOC [Jun et al., 2017]	$\tilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
	OFUGLB [Lee et al., 2024; Liu et al., 2024]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	RS-GLinCB [Sawarni et al., 2024]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}((\log t)^2)^\dagger$	$\mathcal{O}(t)$
	GLB-OMD [**Z**-Xu-Zhao-Sugiyama, 2025]	$\tilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$



- $\kappa = \max_{\mathbf{x}\in\mathcal{X}} 1/\dot{\sigma}(\mathbf{x}^\top \mathbf{w}_*)$
- $\kappa_* = 1/\dot{\sigma}(\mathbf{x}_*^\top \mathbf{w}_*)$ is 1 over the slope at the optimal arm $\mathbf{x}_* = \arg\max_{\mathbf{x}\in\mathcal{X}} \sigma(\mathbf{x}^\top \mathbf{w}_*)$.

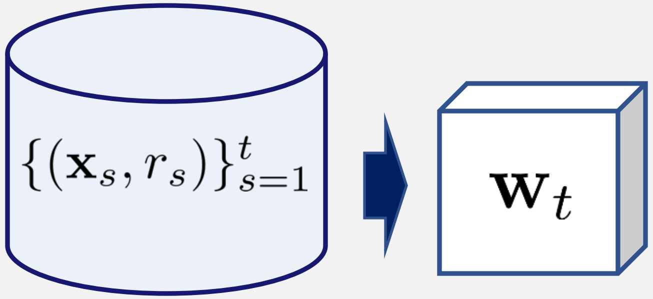
- GLB is almost as efficient as linear bandits.
- **Logistic bandits:** improves upon the best-known existing approach.
- **Unbounded rewards:** applies to Poisson bandits whose rewards are unbounded.

Outline

- Logistic Bandits Problem
- Statistical and Computational Efficient Concern
- Our Jointly Efficient Method
- Extension to Logistic Function Approximation

OFU For Logistic Bandits

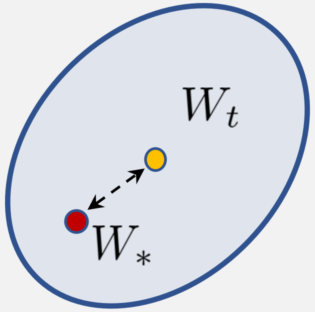
Step 1: Parameter Estimation



historical data

estimator

Step 2: construct high confidence region



$$\|\mathbf{w}_* - \widehat{\mathbf{w}}_t\|_{V_t} \leq \beta_t(\delta)$$

Step 3: select the arm

- For each arm, construct UCB

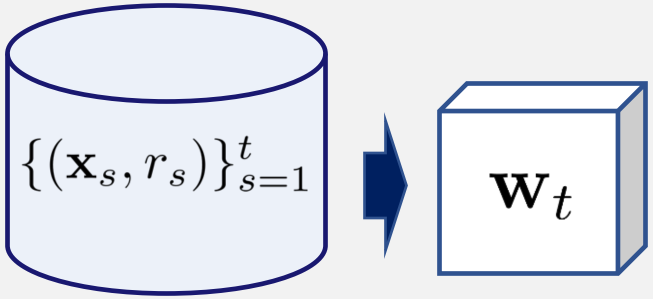
$$\mathrm{UCB}_t(\mathbf{x}) = \max_{W \in \mathcal{C}_t(\delta)} \sigma(\mathbf{w}^\top \mathbf{x})$$

- Select the one with highest UCB

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \mathrm{UCB}_t(\mathbf{x})$$

OFU For Logistic Bandits

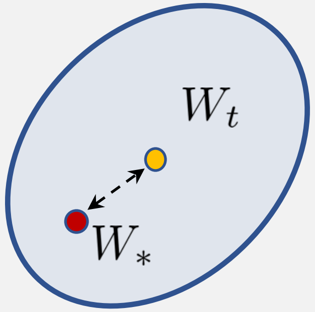
Step 1: Parameter Estimation



historical data

estimator

Step 2: construct high confidence region



$$\|\mathbf{w}_* - \widehat{\mathbf{w}}_t\|_{V_t} \leq \beta_t(\delta)$$

Step 3: select the arm

- For each arm, construct UCB

$$\mathrm{UCB}_t(\mathbf{x}) = \max_{W \in \mathcal{C}_t(\delta)} \sigma(\mathbf{w}^\top \mathbf{x})$$

- Select the one with highest UCB

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} \mathrm{UCB}_t(\mathbf{x})$$

The regret scales with the width of the confidence set $\mathrm{Reg}_T \propto \beta_T(\delta)$

Why κ appears?

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *maximum likelihood estimation* (MLE)

$$\mathbf{w}_t^{\mathrm{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Why κ appears?

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *maximum likelihood estimation* (MLE)

$$\mathbf{w}_t^{\text{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

- κ appears due to improper uncertainty quantification

[Filippi, et al, 2010]: the estimation error of the MLE is proportional to κ

for binary case: $\|\mathbf{w}_* - \mathbf{w}_t^{\text{MLE}}\|_{V_t} \lesssim \kappa\sqrt{d\log T}$

$V_t = \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\top$ is the design matrix

Why κ appears?

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *maximum likelihood estimation* (MLE)

$$\mathbf{w}_t^{\mathrm{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

- κ appears due to improper uncertainty quantification

[Filippi, et al, 2010]: the estimation error of the MLE is proportional to κ

for binary case: $\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} \lesssim \kappa\sqrt{d\log T}$

$V_t = \sum_{s=1}^{t-1} \mathbf{x_s x_s^\top}$ is the design matrix

$$\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} = \left\|\left(\sum_{s=1}^{t-1} \dot{\sigma}\left(\mathbf{x}_s^\top \xi_s\right)\mathbf{x}_s\mathbf{x}_s^\top\right)^{-1} \cdot \left(\sum_{s=1}^{t-1} \epsilon_s \mathbf{x}_s\right)\right\|_{V_t}$$

Why κ appears?

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *maximum likelihood estimation* (MLE)

$$\mathbf{w}_t^{\mathrm{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

- κ appears due to improper uncertainty quantification

[Filippi, et al, 2010]: the estimation error of the MLE is proportional to κ

for binary case: $\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} \leq \kappa\sqrt{d\log T}$

$V_t = \sum^{t-1} \mathbf{x}_s\mathbf{x}_s^\top$ is the design matrix

The same closed-form solution as the least squares, except for the non-linear term

$$\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} = \left\| \left(\sum_{s=1}^{t-1} \dot\sigma\left(\mathbf{x}_s^\top \xi_s\right)\mathbf{x}_s\mathbf{x}_s^\top\right)^{-1} \cdot \left(\sum_{s=1}^{t-1}\epsilon_s\mathbf{x}_s\right)\right\|_{V_t}$$

Why κ appears?

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *maximum likelihood estimation* (MLE)

$$\mathbf{w}_t^{\mathrm{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

- κ appears due to improper uncertainty quantification

[Filippi, et al, 2010]: the estimation error of the MLE is proportional to κ

for binary case: $\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} \lesssim \kappa\sqrt{d\log T}$

$V_t = \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\top$ is the design matrix

$$\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} = \left\|\left(\sum_{s=1}^{t-1} \dot{\sigma}\left(\mathbf{x}_s^\top \xi_s\right)\mathbf{x}_s\mathbf{x}_s^\top\right)^{-1} \cdot \left(\sum_{s=1}^{t-1} \epsilon_s \mathbf{x}_s\right)\right\|_{V_t} \leq \kappa \left\|\left(\sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\top\right)^{-1} \cdot \left(\sum_{s=1}^{t-1} \epsilon_s \mathbf{x}_s\right)\right\|_{V_t} \leq \mathcal{O}\left(\kappa\sqrt{d\log T}\right)$$

Why κ appears? → the local non-linearity of MLE is not taken into account.

Why κ appears?

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *maximum likelihood estimation* (MLE)

$$\mathbf{w}_t^{\mathrm{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

- κ appears due to improper uncertainty quantification

[Filippi, et al, 2010]: the estimation error of the MLE is proportional to κ

for binary case: $\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{V_t} \lesssim \kappa\sqrt{d\log T}$

$V_t = \sum_{s=1}^{t-1} \mathbf{x_s x_s^\top}$ is the design matrix

[Faury, et al, 2020]: capture the local curvature of the MLE estimator

approximate $\dot\sigma(\mathbf{x}_s^\top \xi_s)$ by the term $\dot\sigma(\mathbf{x}_s^\top \mathbf{w}_*)$

$$\|\mathbf{w}_* - \mathbf{w}_t^{\mathrm{MLE}}\|_{H_t(\mathbf{w}_*)} \lesssim \sqrt{d\log T}$$

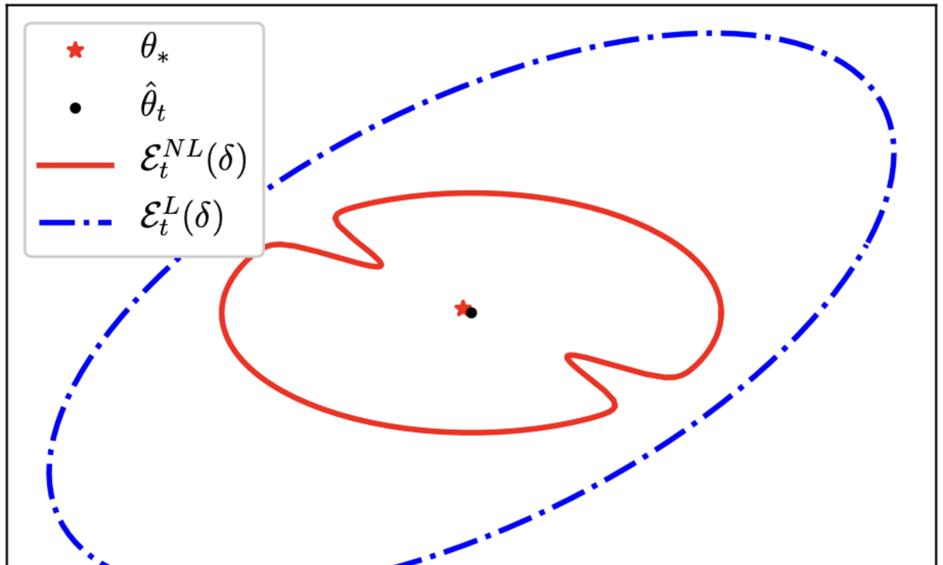
$$H_t(\mathbf{w}) = \sum_{S=1}^{t-1} \dot\sigma\left(\mathbf{x}_s^\top \mathbf{w}_*\right) \mathbf{x}_s \mathbf{x}_s^\top$$

Why κ appears

- **Parameter Estimation**: estimate the $\mathbf{w}_*$ by *m*

$$\mathbf{w}_t^{\text{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w})$$

- κ appears due to improper uncertainty quant



[Faury et al., 2020 Figure 2]

[Filippi, et al, 2010]: the estimation error of the MLE is pro

for binary case: $\|\mathbf{w}_* - \mathbf{w}_t^{\text{MLE}}\|_{V_t} \lesssim \kappa\sqrt{d\log T}$

$V_t = \sum_{s=1}^{t-1} \mathbf{x_s}\mathbf{x_s}^\top$ is the design matrix

[Faury, et al, 2020]: capture the local curvature of the MLE estimator

approximate $\dot{\sigma}(\mathbf{x}_s^\top \xi_s)$ by the term $\dot{\sigma}(\mathbf{x}_s^\top \mathbf{w}_*)$

$$\|\mathbf{w}_* - \mathbf{w}_t^{\text{MLE}}\|_{H_t(\mathbf{w}_*)} \lesssim \sqrt{d\log T}$$

$$H_t(\mathbf{w}) = \sum_{S=1}^{t-1} \dot{\sigma}\left(\mathbf{x}_s^\top \mathbf{w}_*\right) \mathbf{x}_s \mathbf{x}_s^\top$$

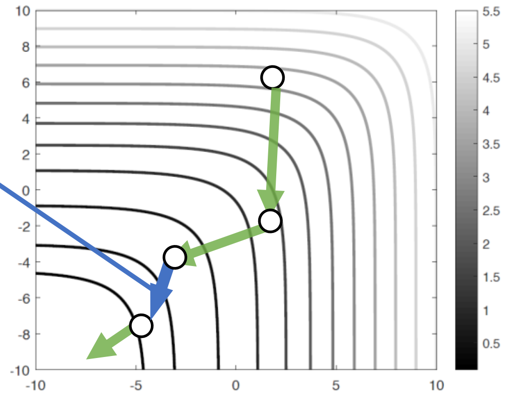
Computational Concern

- *Maximum likelihood estimation is computationally inefficient*

$$\mathbf{w}_t^{\text{MLE}} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{s=1}^{t-1} \ell_s(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Per gradient descent step:

- $\mathcal{O}(t)$ time complexity per step
- $\mathcal{O}(t)$ storage complexity per step



Our solution

- ***Online Estimator:*** learn the parameter with the online mirror descent

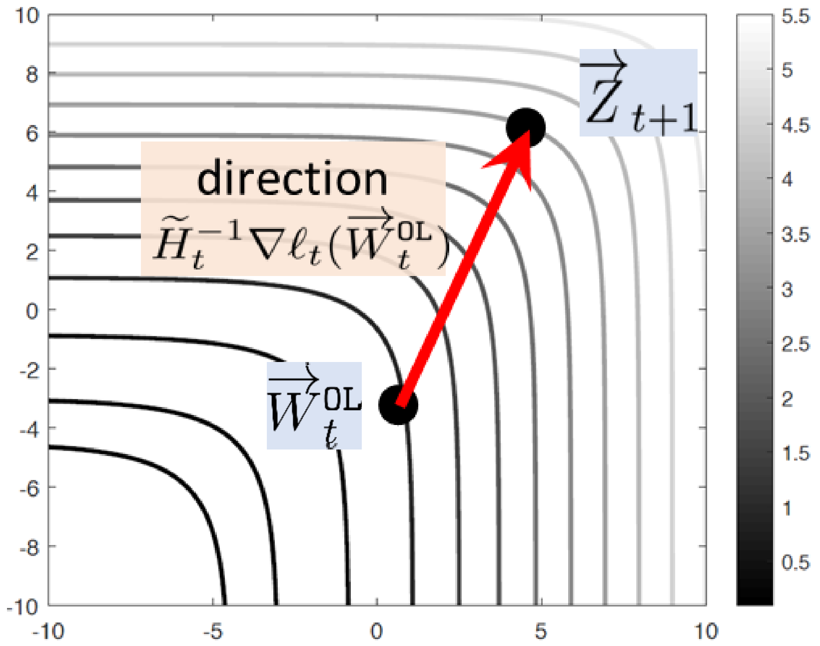
$$\mathbf{z}_{t+1} = \mathbf{w}_t - \eta \widetilde{H}_t^{-1} \nabla \ell_t(\mathbf{w}_t)$$

gradient update step

- $\eta > 0$ is the ste

$\mathbf{w}_t$ is used to approximate $\mathbf{w}_*$ and it is sufficient

- $\widetilde{H}_t = \lambda I + \sum_{s=1}^{t-1} \dot{\sigma}(\mathbf{w}_{s+1}^\top \mathbf{x}_s)\mathbf{x}_s\mathbf{x}_s^\top + \eta \dot{\sigma}(\mathbf{w}_t^\top \mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^\top$ is a carefully designed matrix to exploit the local curvature of the loss function.

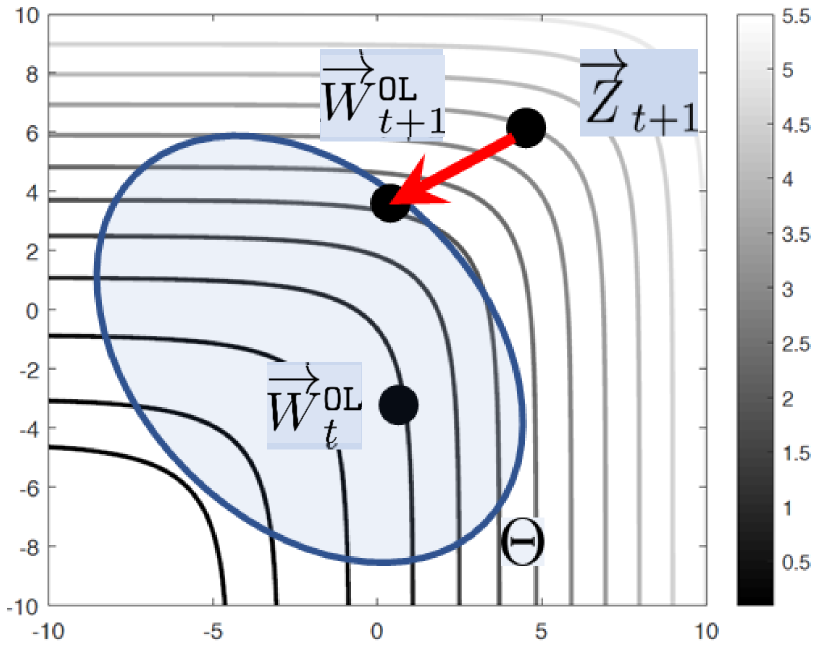


Our solution

- ***Online Estimator:*** learn the parameter with the online mirror descent

$$\mathbf{w}^{\mathtt{OL}}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}} \|\mathbf{w} - \mathbf{z}_{t+1}\|_{\widetilde{H}_t},$$

Projection step



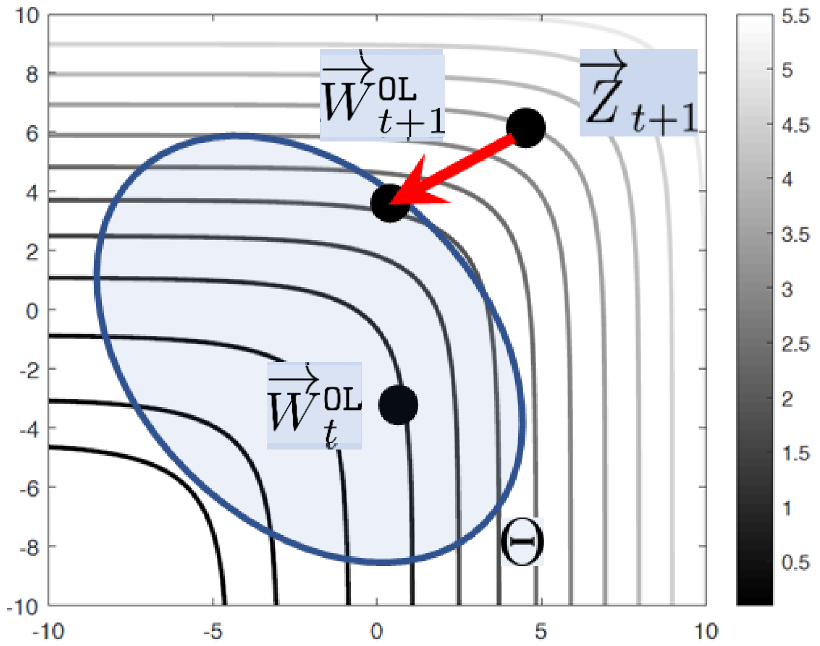
Our solution

- ***Online Estimator:*** learn the parameter with the online mirror descent

$$\mathbf{w}^{\mathtt{OL}}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}} \|\mathbf{w} - \mathbf{z}_{t+1}\|_{\widetilde{H}_t},$$

Projection step

Our method is free from storing all historical data



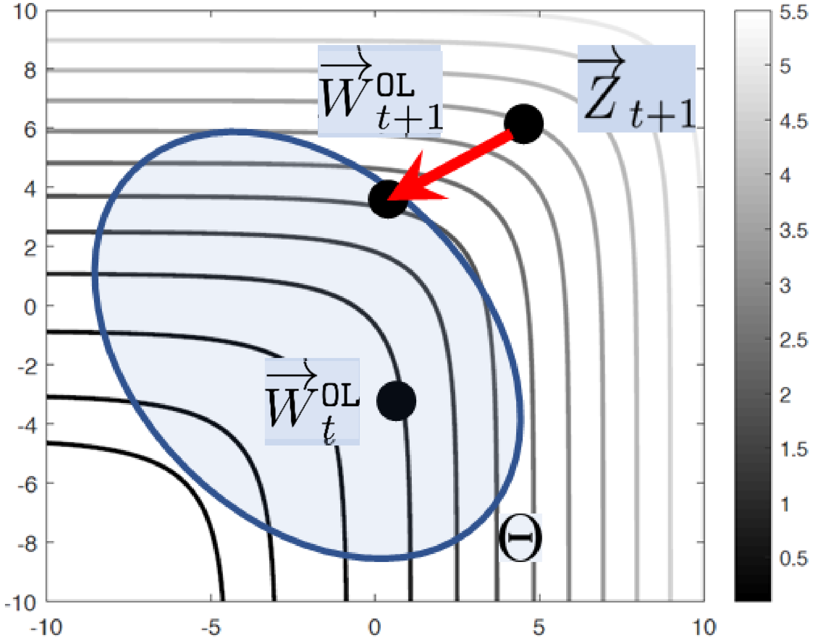
Our solution

- ***Online Estimator:*** learn the parameter with the online mirror descent

$$\mathbf{w}_{t+1}^{\mathtt{OL}} = \arg\min_{\mathbf{w}\in\mathcal{W}} \|\mathbf{w} - \mathbf{z}_{t+1}\|_{\widetilde{H}_t},$$

Projection step

Our method is free from storing all historical data



How are the statistical properties? Any loss?

Our solution

Main Theorem (informal): *With appropriate configuration of the step size* η *and regularization coefficient* λ, *for each iteration* $t \in [T]$,

Independent of κ

$$\left\| \mathbf{w}_t^{\mathtt{OL}} - \mathbf{w}_* \right\|_{H_t} \lesssim \sqrt{d \log t},$$

where $\mathbf{w}_t^{\mathtt{OL}}$ *is the online estimator and* $H_t = \lambda I + \sum_{s=1}^{t-1} \dot{\sigma}(\mathbf{w}_{s+1}^\top \mathbf{x}_s) \mathbf{x}_s \mathbf{x}_s^\top$.

Our solution

***Main Theorem (informal)**: With appropriate configuration of the step size η and regularization coefficient λ, for each iteration $t \in [T]$, we have*

$$\left\| \mathbf{w}_t^{\mathtt{OL}} - \mathbf{w}_* \right\|_{H_t} \lesssim \sqrt{d \log t},$$

where $\mathbf{w}_t^{\mathtt{OL}}$ *is the online estimator and* $H_t = \lambda I + \sum_{s=1}^{t-1} \dot{\sigma}(\mathbf{w}_{s+1}^\top \mathbf{x}_s) \mathbf{x}_s \mathbf{x}_s^\top$.

- Jointly efficient estimator **for multinomial logistic regression**:
  - Computationally efficient: $\mathcal{O}(1)$ computational and storage cost per round
  - Statistically efficient: "κ-independent" estimation error

Our solution

Main Theorem (informal): *With appropriate configuration of the step size η and regularization coefficient λ, for each iteration $t \in [T]$, we have*

$$\left\|\mathbf{w}_t^{\mathtt{OL}} - \mathbf{w}_*\right\|_{H_t} \lesssim \sqrt{d \log t},$$

where $\mathbf{w}_t^{\mathtt{OL}}$ *is the online estimator and* $H_t = \lambda I + \sum_{s=1}^{t-1} \dot{\sigma}(\mathbf{w}_{s+1}^\top \mathbf{x}_s)\mathbf{x}_s\mathbf{x}_s^\top$.

- Jointly efficient estimator **for multinomial logistic regression**:
  - Computationally efficient: $\mathcal{O}(1)$ computational and storage cost per round
  - Statistically efficient: "κ-independent" estimation error

$$\mathcal{C}_t^{\mathtt{OL}}(\delta) \triangleq \left\{\mathbf{w} \in \mathcal{X} \mid \left\|\mathbf{w}_t^{\mathtt{OL}} - \mathbf{w}\right\|_{H_t} \lesssim \sqrt{d \log t}\right\}$$

ellipsoid confidence set to construct UCB

Joint Efficient Algorithm

Algorithm 1 GLB-OMD

1: **Input:** regularization coefficient λ, probability δ, step size η.
2: Initialize $H_1 = \lambda I_{Kd}$ and $\overrightarrow{W}_1^{\mathtt{OL}}$ as any point in $\mathcal{W}$
3: **for** $t = 1, \ldots, T$ **do**
4: Select the arm by $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{X}} \mathtt{UCB}_t(\mathbf{x})$ and receive y_t.
5: Update $\tilde{H}_t = H_t + \eta\mu'(\mathbf{w}_t^{\mathtt{OL}}\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^\top$
6: Update the estimator $\mathbf{w}_{t+1}^{\mathtt{OL}}$ for the next iteration by (6)
7: Update $H_{t+1} = H_t + \mu'(\mathbf{w}_{t+1}^{\mathtt{OL}}\mathbf{x}_t)\mathbf{x}_t\mathbf{x}_t^\top$ and
8: Construct UCB by $\mathtt{UCB}_{t+1}(\mathbf{x}) = \arg\max_{\mathbf{w}\in\mathcal{C}_{t+1}(\delta)} \mu(\mathbf{x}^\top\mathbf{w})$.
9: **end for**

online update of the estimator

construct UCB with an ellipsoid

***Theorem 2**: With appropriate configuration of the step size η and regularization coefficient λ, for each iteration t ∈ [T], we have*

$$\mathrm{Reg}_T \lesssim d\log T\sqrt{\frac{T}{\kappa_*}} + \kappa d^2(\log T)^2$$

Joint Efficient Algorithm

Algorithm 1 GLB-OMD

Input: regularization coefficient λ, probability δ, step size η.
Initialize $H_1 = \lambda I_{Kd}$ and $\overrightarrow{W}_1^{\mathrm{OL}}$ as any point in $\mathcal{W}$
for $t = 1, \ldots, T$ **do**
  Select the arm by $\mathbf{x}_t = \arg\max_{\mathbf{x} \in \mathcal{X}} \mathtt{UCB}_t(\mathbf{x})$ and receive y_t.
  Update $\tilde{H}_t = H_t + \eta \mu'(\mathbf{w}_t^{\mathtt{OL}} \mathbf{x}_t) \mathbf{x}_t \mathbf{x}_t^\top$
  Update the estimator $\mathbf{w}_{t+1}^{\mathtt{OL}}$ for the next iteration by (6)
  Update $H_{t+1} = H_t + \mu'(\mathbf{w}_{t+1}^{\mathtt{OL}} \mathbf{x}_t) \mathbf{x}_t \mathbf{x}_t^\top$ and

online update of the estimator

construct UCB with

best-known $\tilde{\mathcal{O}}(T/\kappa_*)$ regret bound with $\mathcal{O}(1)$ cost per round

***Theorem 2**: With appropriate configuration of the step size η and regularization coefficient λ, for each iteration t ∈ [T], we have*

$$\mathrm{Reg}_T \lesssim d \log T \sqrt{\frac{T}{\kappa_*}} + \kappa d^2 (\log T)^2$$

Summary & Future Work

- For generalized linear bandits, a single gradient step is enough to ensure statistical efficiency

Setting	Algorithm	Regret	Comput. per Round	Storage Cost
linear	OFUL [Abbasi-Yadkori et al., 2011]	$\widetilde{\mathcal{O}}(\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
generalized linear (GLB)	GLM-UCB [Filippi et al., 2010]	$\widetilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	GLOC [Jun et al., 2017]	$\widetilde{\mathcal{O}}(\kappa\sqrt{T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
	OFUGLB [Lee et al., 2024; Liu et al., 2024]	$\widetilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(t)$	$\mathcal{O}(t)$
	RS-GLinCB [Sawarni et al., 2024]	$\widetilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}((\log t)^2)^\dagger$	$\mathcal{O}(t)$
	GLB-OMD [**Z**-Xu-Zhao-Sugiyama, 2025]	$\widetilde{\mathcal{O}}(\sqrt{T/\kappa_*})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Future questions:

- totally free of kappa?
- beyond the linear class

- More potentials: dueling bandits, RLHF, Function Approximation…

Thanks!
Q&A