

Batched Nonparametric Contextual Bandits



Cong Ma

Department of Statistics, UChicago

Wilks Seminar, Princeton ORFE, Apr. 2024



Rong Jiang
UChicago CCAM

Multi-armed bandits

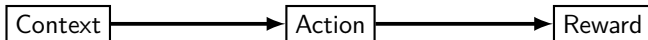
— Robbins, 1952, Lai and Robbins, 1985



- sequential decision making
- time horizon T
- action set: K arms
- random reward for each action
- goal: maximize expected cumulative reward

Multi-armed bandits with covariates (aka contextual bandits)

— Yang and Zhu, 2002, Rigollet and Zeevi, 2010, Perchet and Rigollet, 2013



Contextual bandit is widely used for personalized decision making, e.g.

- online recommender system
- personalized digital health

Batch constraints

— Perchet et al., 2016, Gao et al., 2019, Fan et al., 2023

- Data may arrive in batches
- Rewards are observed at the end of a batch
- Statistician cannot update the policy too frequently, especially when the number of users are large
- Common in a lot of applications including clinical trials and online recommendation

Main questions

- What's the optimal way to select batch sizes, and to update policy after each batch?
- Is it possible to achieve regret performances as in the fully online setting using few policy updates?

Problem setup

A 2-armed nonparametric bandit is specified by a sequence of iid tuples

$$\{(X_t, Y_t^{(1)}, Y_t^{(-1)})\}_{1 \leq t \leq T}$$

- T is time horizon
- Context $X_t \in \mathcal{X} = [0, 1]^d$ follows distribution P_X
- Reward $Y_t^{(k)} \in [0, 1]$ with $\mathbb{E}[Y_t^{(k)} \mid X_t] = f^{(k)}(X_t)$ for arm $k \in \{1, -1\}$. Call $f^{(k)}$ reward function of arm k

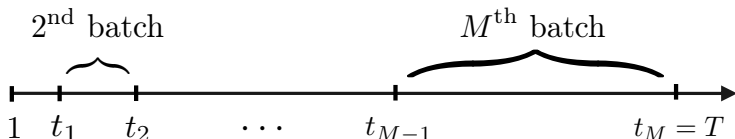
Game rules w/o batch constraints

The game is sequential: at each step t , statistician

- observes context X_t
- selects action A_t according to rule $\pi_t : \mathcal{X} \mapsto \{1, -1\}$
- and then receives corresponding reward $Y_t^{(A_t)}$

Key: π_t is allowed to depend on all observations prior to step t

Game rules with batch constraints



Given M —number of allowed batches, statistician needs to decide on M -batch policy (Γ, π) :

- $\Gamma = \{t_1, \dots, t_M = T\}$ is a partition of the entire time horizon T
- $\pi = \{\pi_t\}_{t=1}^T$, where $\pi_t : \mathcal{X} \mapsto \{1, -1\}$
- π_t only depends on all observations prior to current batch

Regret minimization

Define optimal reward function $f^*(x) = \max_{k \in \{1, -1\}} f^{(k)}(x)$

Goal: minimize expected regret

$$R_T(\pi) = \mathbb{E} \left[\sum_{t=1}^T \left(f^*(X_t) - f^{(\pi_t(X_t))}(X_t) \right) \right]$$

Problem assumptions

- **Smoothness.** There exist $\beta \in (0, 1]$ and $L > 0$ such that

$$|f^{(k)}(x) - f^{(k)}(x')| \leq L\|x - x'\|_2^\beta,$$

for $k \in \{1, -1\}$ and $x, x' \in \mathcal{X}$

- **Margin.** There exist $\alpha > 0$, $\delta_0 \in (0, 1)$ and $D_0 > 0$ such that

$$\mathbb{P}_X \left(0 < \left| f^{(1)}(X) - f^{(-1)}(X) \right| \leq \delta \right) \leq D_0 \delta^\alpha$$

holds for all $\delta \in [0, \delta_0]$

Problem assumptions

- **Smoothness.** There exist $\beta \in (0, 1]$ and $L > 0$ such that

$$|f^{(k)}(x) - f^{(k)}(x')| \leq L \|x - x'\|_2^\beta,$$

for $k \in \{1, -1\}$ and $x, x' \in \mathcal{X}$

- **Margin.** There exist $\alpha > 0$, $\delta_0 \in (0, 1)$ and $D_0 > 0$ such that

$$\mathbb{P}_X \left(0 < \left| f^{(1)}(X) - f^{(-1)}(X) \right| \leq \delta \right) \leq D_0 \delta^\alpha$$

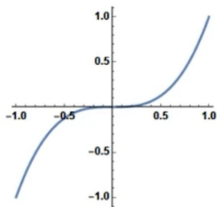
holds for all $\delta \in [0, \delta_0]$

→ Problem class $\mathcal{F}_{\alpha, \beta}$

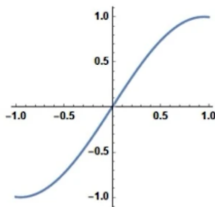
Margin conditions

Margin condition:

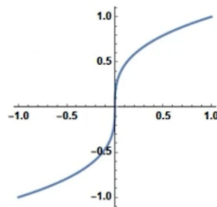
$$\mathbb{P}_X \left(0 < \left| f^{(1)}(X) - f^{(-1)}(X) \right| \leq \delta \right) \leq D_0 \delta^\alpha$$



$\alpha < 1$



$\alpha = 1$



$\alpha > 1$

borrowed from Nathan Kallus's slides

Interesting regime

We only focus on $\alpha\beta \leq 1$ since

- When $\alpha\beta > 1$, contexts do not matter: there exists a single arm that is uniformly optimal
- When $\alpha\beta \leq 1$, there exists nontrivial contextual bandits in $\mathcal{F}_{\alpha,\beta}$

Prior work

Define $\gamma := \frac{\beta(1+\alpha)}{2\beta+d}$

Theorem 0 (Rigollet and Zeevi, '10, Perchet and Rigollet, '13)

In fully online setting, i.e., $M = T$, we have

$$\inf_{(\Gamma, \pi)} \sup_{\mathcal{F}_{\alpha, \beta}} \mathbb{E}[R_T(\pi)] \asymp T^{1-\gamma}$$

Our results

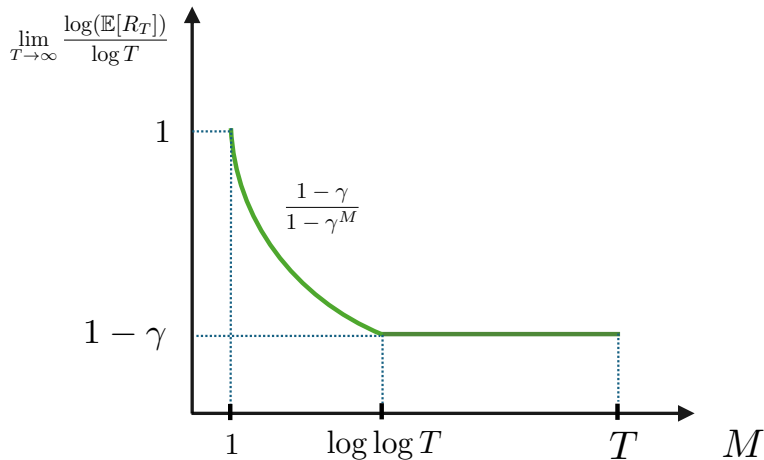
Recall $\gamma = \frac{\beta(1+\alpha)}{2\beta+d}$

Theorem 1 (Jiang, Ma, 2024)

Fix M , number of batches. We have, up to log factors,

$$\inf_{(\Gamma, \pi)} \sup_{\mathcal{F}_{\alpha, \beta}} \mathbb{E}[R_T(\pi)] \asymp \begin{cases} T^{\frac{1-\gamma}{1-\gamma}M}, & \text{when } M \lesssim \log \log T, \\ T^{1-\gamma}, & \text{when } M \gtrsim \log \log T \end{cases}$$

Our results in a figure



Minimax lower bounds

Theorem 2 (Jiang and Ma, 2024)

Assume P_X is the uniform distribution on \mathcal{X} . Any M -batch policy (Γ, π) has worst-case regret

$$\mathbb{E}[R_T(\pi)] \gtrsim T^{\frac{1-\gamma}{1-\gamma M}}$$

- this together with lower bound for $M = T$ in Rigollet and Zeevi, 2010 leads to our final lower bounds

Minimax upper bounds

Theorem 3 (Jiang and Ma, 2024)

Assume $M = O(\log T)$. Algorithm BaSEDB (to be introduced) achieves

$$\mathbb{E}[R_T(\hat{\pi})] \lesssim (\log T)^2 \cdot T^{\frac{1-\gamma}{1-\gamma M}}$$

- An immediate implication: when $M \gtrsim \log \log T$, BaSEDB achieves optimal regret $T^{1-\gamma}$ in fully online setting

Analysis for lower bounds

and why it is instrumental

Notations

- M -batch policy (Γ, π) with

$$\Gamma = \{t_1, t_2, \dots, t_M = T\}$$

- Bernoulli rewards: $Y_t^{(1)}, Y_t^{(-1)}$ are Bernoulli random variables with mean $f^{(1)}(X_t)$, and $f^{(-1)}(X_t)$, respectively
- Fix $f^{(-1)}(x) = \frac{1}{2}$, and denote by f be the mean reward function of arm 1
- Cumulative regret up to time t : $R_t(\pi; f)$

Notations

- M -batch policy (Γ, π) with

$$\Gamma = \{t_1, t_2, \dots, t_M = T\}$$

- Bernoulli rewards: $Y_t^{(1)}, Y_t^{(-1)}$ are Bernoulli random variables with mean $f^{(1)}(X_t)$, and $f^{(-1)}(X_t)$, respectively
- Fix $f^{(-1)}(x) = \frac{1}{2}$, and denote by f be the mean reward function of arm 1
- Cumulative regret up to time t : $R_t(\pi; f)$

Target: lower bound $\sup_{(f, \frac{1}{2}) \in \mathcal{F}(\alpha, \beta)} R_T(\pi; f)$

A simple but key observation

Worst-case regret over $[T]$ is larger than that over first i batches

Formally, we have

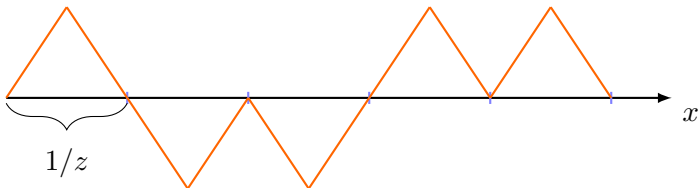
$$\sup_{(f, \frac{1}{2}) \in \mathcal{F}(\alpha, \beta)} R_T(\pi; f) \geq \max_{1 \leq i \leq M} \sup_{(f, \frac{1}{2}) \in \mathcal{F}(\alpha, \beta)} R_{t_i}(\pi; f)$$

Though simple, this observation lends us freedom on choosing different families of instances in $\mathcal{F}(\alpha, \beta)$ targeting different batch indices i

Family of reward instances

— long history in nonparametric estimation

- Split $[0, 1]$ to z number of equal-sized bins
- Place a random hat function on each bin



Worst-case regret over $[t_i]$

Set $z = z_i = \lceil (t_{i-1})^{1/(2\beta+d)} \rceil$.

By standard calculations, we obtain

$$\sup_{(f, \frac{1}{2}) \in \mathcal{F}(\alpha, \beta)} R_{t_i}(\pi; f) \gtrsim \begin{cases} \frac{t_i}{t_{i-1}^\gamma}, & i > 1 \\ t_1, & i = 1 \end{cases}$$

Putting things together

$$\begin{aligned} \sup_{(f, \frac{1}{2}) \in \mathcal{F}(\alpha, \beta)} R_T(\pi; f) &\geq \max_{1 \leq i \leq M} \sup_{(f, \frac{1}{2}) \in \mathcal{F}(\alpha, \beta)} R_{t_i}(\pi; f) \\ &\geq \max_{1 \leq i \leq M} \sup_{f \in \mathcal{C}_{z_i}} R_{t_i}(\pi; f) \\ &\gtrsim \max \left\{ t_1, \frac{t_2}{t_1^\gamma}, \dots, \frac{T}{t_{M-1}^\gamma} \right\} \\ &\asymp T^{\frac{1-\gamma}{1-\gamma M}} \end{aligned}$$

Implications on optimal M -batch policy

- **Grid selection:** in view of lower bound

$$\max \left\{ t_1, \frac{t_2}{t_1^\gamma}, \dots, \frac{T}{t_{M-1}^\gamma} \right\},$$

one needs to set

$$t_1 \asymp \frac{t_i}{t_{i-1}^\gamma} \asymp T^{\frac{1-\gamma}{1-\gamma M}} \quad \text{for } 2 \leq i \leq M$$

Any other choice of $\Gamma = \{t_1, t_2, \dots, t_M = T\}$ has higher worst-case regret

Implications on optimal M -batch policy

- **Dynamic binning:** recall for each different batch i , we set

$$z = z_i = \lceil t_{i-1}^{1/(2\beta+d)} \rceil$$

In other words, the granularity (i.e., bin width $1/z_i$) at which we investigate mean reward functions depends crucially on grid points $\{t_i\}$: the larger the grid point t_i , the finer the granularity

Implications on optimal M -batch policy

- **Dynamic binning:** recall for each different batch i , we set

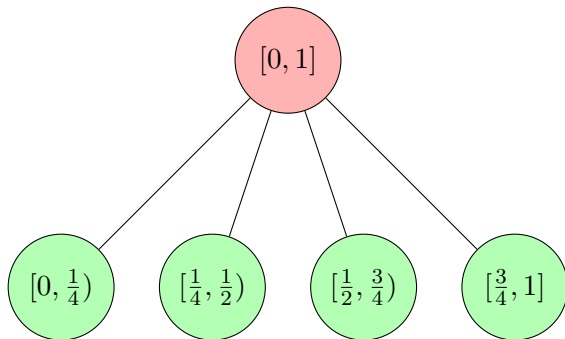
$$z = z_i = \lceil t_{i-1}^{1/(2\beta+d)} \rceil$$

In other words, the granularity (i.e., bin width $1/z_i$) at which we investigate mean reward functions depends crucially on grid points $\{t_i\}$: the larger the grid point t_i , the finer the granularity

⇒ batched successive elimination with dynamic binning algorithm

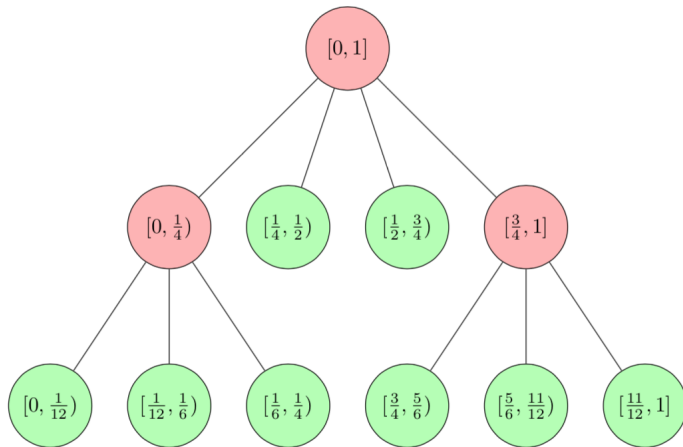
Batched successive elimination with dynamic binning

Prior to 1st batch:



Batched successive elimination with dynamic binning

After 1st batch (or prior to 2nd batch):



A tree-based interpretation

- \mathcal{L} is a list of active bins, and \mathcal{I}_C is the active arms for bin C
- Prior to batch 1: $\mathcal{L} \leftarrow \mathcal{B}_1$, where \mathcal{B}_1 is a regular partition of \mathcal{X} with bins of equal width w_1 . In the above example, $w_1 = 1/4$
- Within this batch: try the arms in \mathcal{I}_C equally likely whenever a sample $X_t \in C$
- At the end of the batch: given the revealed rewards, update \mathcal{I}_C for each $C \in \mathcal{L}$ via successive elimination
- If no arm were eliminated from \mathcal{I}_C , split the bin $C \in \mathcal{L}$ into its children $\text{child}(C)$ and replace C with $\text{child}(C)$
- Repeat the above process in a batch fashion

Batched successive elimination with dynamic binning

- Input: Batch size M , grid $\Gamma = \{t_i\}_{i=0}^M$, split factors $\{g_i\}_{i=0}^{M-1}$
- $\mathcal{L} \leftarrow \mathcal{B}_1$
- $\mathcal{I}_C \leftarrow \{1, -1\}$ for $C \in \mathcal{L}$
- for $i = 1, \dots, M - 1$ do
 - for $t = t_{i-1} + 1, \dots, t_i$ do
 - ① $C \leftarrow \mathcal{L}(X_t)$
 - ② Pull an arm from \mathcal{I}_C in a round-robin way
 - ③ Update \mathcal{L} and $\{\mathcal{I}_C\}_{C \in \mathcal{L}}$ when $t = t_i$
- for $t = t_{M-1} + 1, \dots, T$ do
 - ① $C \leftarrow \mathcal{L}(X_t)$
 - ② Pull any arm from \mathcal{I}_C

Performance guarantees

Denote $b \asymp T^{\frac{1-\gamma}{1-\gamma M}}$. Choose

- batch sizes

$$t_1 \asymp T^{\frac{1-\gamma}{1-\gamma M}}, \quad \text{and} \quad t_i = \lfloor b(t_{i-1})^\gamma \rfloor, \quad \text{for } i = 2, \dots, M$$

- split factors

$$g_0 = \lfloor b^{\frac{1}{2\beta+d}} \rfloor, \quad \text{and} \quad g_i = \lfloor g_{i-1}^\gamma \rfloor, i = 1, \dots, M-2$$

Theorem 3 (More explicit version)

When $M = O(\log T)$, BaSEDB with above choices of batch sizes and split factors obeys

$$\mathbb{E}[R_T(\hat{\pi})] \lesssim (\log T)^2 \cdot T^{\frac{1-\gamma}{1-\gamma M}}$$

Numerics

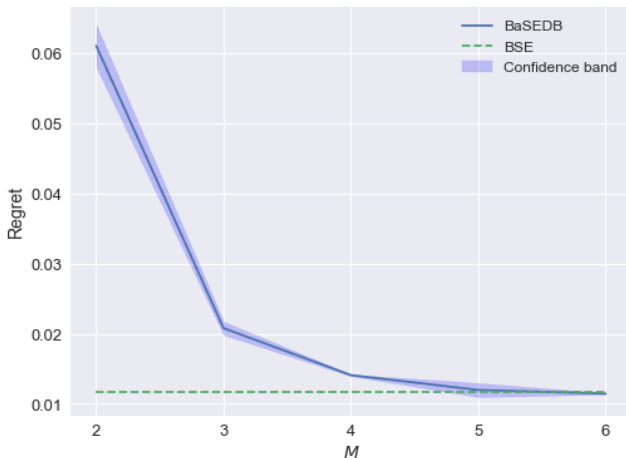


Figure 1: Default parameters are $T = 50000, d = 1, \alpha = 0.2, \beta = 1$. BSE is regret optimal policy without the batch constraint.

Is dynamic binning necessary?

- Without batch constraint, successive elimination with static binning achieves optimal regret
- We motivate dynamic binning by proof of lower bounds; but maybe we are not smart enough to find a single family of instances that are hard for all batches

Suboptimality of static binning

Theorem 4 (Jiang and Ma, 2024)

Consider $\alpha = \beta = d = 1$ and $M = 3$. For any 3-batch SE policy (Γ, π) with a fixed number of g bins. No matter how one sets g , there exists a nonparametric bandit instance such that

$$\mathbb{E}[R_T(\hat{\pi})] \gg T^{\frac{9}{19}},$$

where $T^{\frac{9}{19}}$ is the optimal regret achieved by BaSEDB

This demonstrates the necessity of dynamic binning in some sense

Concluding remarks

Summary:

- Batched successive elimination with dynamic binning is nearly minimax optimal
- It is almost necessary as static binning is strictly suboptimal

Concluding remarks

Summary:

- Batched successive elimination with dynamic binning is nearly minimax optimal
- It is almost necessary as static binning is strictly suboptimal

Future directions:

- Remove log factors
- Adaptive to margin parameters
- static grid vs. adaptive grid

Concluding remarks

Summary:

- Batched successive elimination with dynamic binning is nearly minimax optimal
- It is almost necessary as static binning is strictly suboptimal

Future directions:

- Remove log factors
- Adaptive to margin parameters
- static grid vs. adaptive grid

Paper:

- R. Jiang, and C. Ma, "Batched nonparametric contextual bandits," arXiv:2402.17732, 2024