Learning with Few Updates: Batched Contextual Bandits

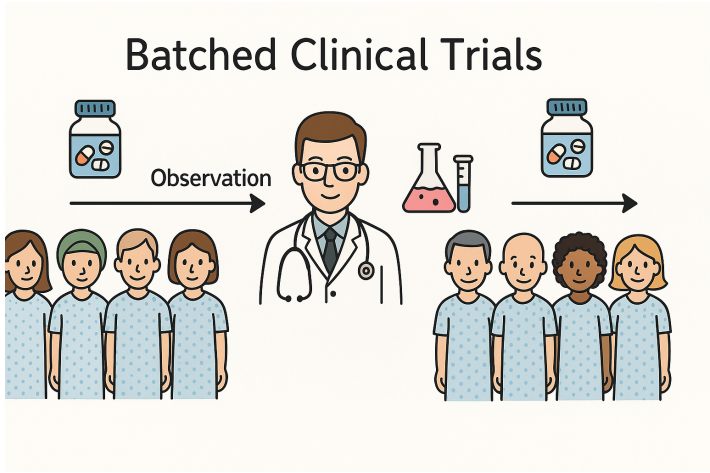
Cong Ma

Department of Statistics, UChicago

Statistics Seminar, UIUC, Nov. 2025

Rong Jiang
UChicago CCAM

Sequential decision-making with batch constraints

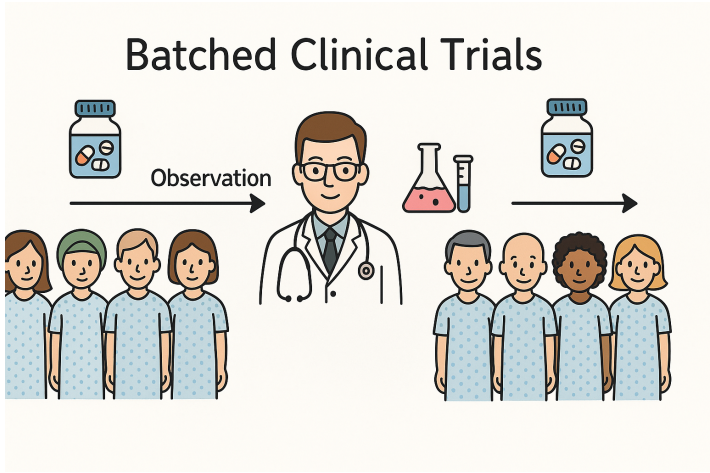


(a) Clinical trials

(b) Online platforms

- Clinical trials: groups of patients treated in stages
- Online platforms: policies are retrained every few hours/days

Sequential decision-making with batch constraints



(a) Clinical trials

(b) Online platforms

- Clinical trials: groups of patients treated in stages
- Online platforms: policies are retrained every few hours/days

How many updates are needed for optimal decision-making?

Roadmap for today

- A simple model: contextual bandits with batch constraints
- Part I: Known margin parameter α
- Part II: Unknown margin α: an adaptivity barrier
- Closing remarks

A simple model:
Contextual bandits with batch constraints

Contextual bandits

At each round $t = 1, 2, \ldots, T$:

- Observe **context** X_t: a patient's features or a user profile
- Choose **action** A_t: which treatment to assign or which recommendation to display
- Observe **reward** $Y_t(A_t) \in [0, 1]$: health outcome or click indicator

Contextual bandits

At each round $t = 1, 2, \ldots, T$:

- Observe **context** X_t: a patient's features or a user profile
- Choose **action** A_t: which treatment to assign or which recommendation to display
- Observe **reward** $Y_t(A_t) \in [0, 1]$: health outcome or click indicator

The expected reward depends on both the action and the context:

$$\mathbb{E}[Y_t(k) \mid X_t] = f^{(k)}(X_t), \qquad k \in \{1, -1\}.$$

Contextual bandits

At each round $t = 1, 2, \ldots, T$:

- Observe **context** X_t: a patient's features or a user profile
- Choose **action** A_t: which treatment to assign or which recommendation to display
- Observe **reward** $Y_t(A_t) \in [0, 1]$: health outcome or click indicator

The expected reward depends on both the action and the context:

$$\mathbb{E}[Y_t(k) \mid X_t] = f^{(k)}(X_t), \qquad k \in \{1, -1\}.$$

Regret minimization

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

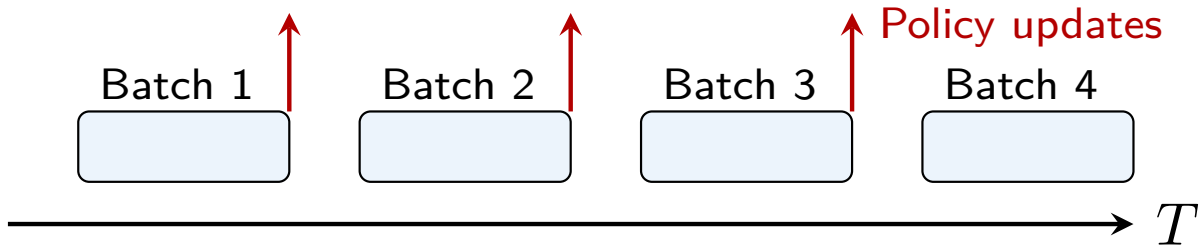
Batch constraint

We have T rounds, but we can only update the policy M times.

Batch constraint

We have T rounds, but we can only update the policy M times.
Statistician needs to decide on M-batch policy (Γ, π):

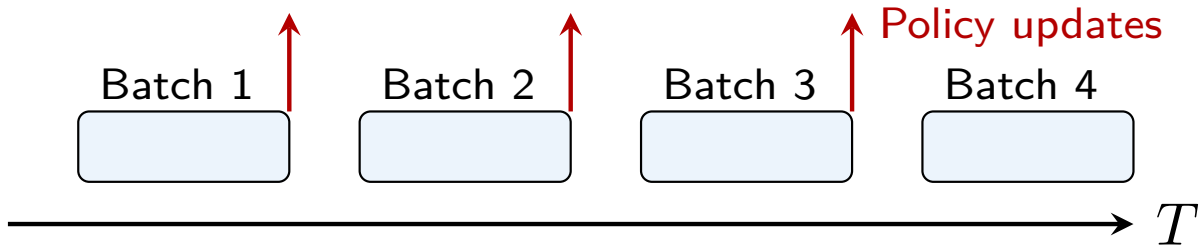
- $\Gamma = \{0 = t_0 < t_1, \ldots, t_{M-1} < t_M = T\}$ is a partition of the entire time horizon T
  - fixed grid vs. adaptive grid
- $\pi = \{\pi_t\}_{1 \le t \le T}$, where $\pi_t : \mathcal{X} \mapsto \{1, -1\}$
- π_t only depends on all observations prior to current batch



Batch constraint

We have T rounds, but we can only update the policy M times.
Statistician needs to decide on M-batch policy (Γ, π):

- $\Gamma = \{0 = t_0 < t_1, \ldots, t_{M-1} < t_M = T\}$ is a partition of the entire time horizon T
  - fixed grid vs. adaptive grid
- $\pi = \{\pi_t\}_{1 \le t \le T}$, where $\pi_t : \mathcal{X} \mapsto \{1, -1\}$
- π_t only depends on all observations prior to current batch



Question: How does regret scale with the number of batches?

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 $x, x' \in \mathcal{X}$

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 $\alpha \geq 0$, $\delta_0 \in (0, 1)$ 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 $\alpha \geq 0$, $\delta_0 \in (0, 1)$ 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 $\delta \in [0, \delta_0]$

Together, (α, β) determine the difficulty of learning $\rightarrow \mathcal{P}_{\alpha,\beta}$

Nontrivial regime $\alpha\beta \leq d$

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

- When $\alpha\beta > d$, contexts do not matter: there exists a single arm that is uniformly optimal
- When $\alpha\beta \leq d$, there exist nontrivial contextual bandits in $\mathcal{P}_{\alpha,\beta}$: optimal policy must be context-dependent

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*

$$\inf_{(\Gamma,\pi)} \sup_{P \in \mathcal{P}_{\alpha,\beta}} R_T(\Gamma, \pi; P) \asymp T^{1-\gamma}$$

Our results

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

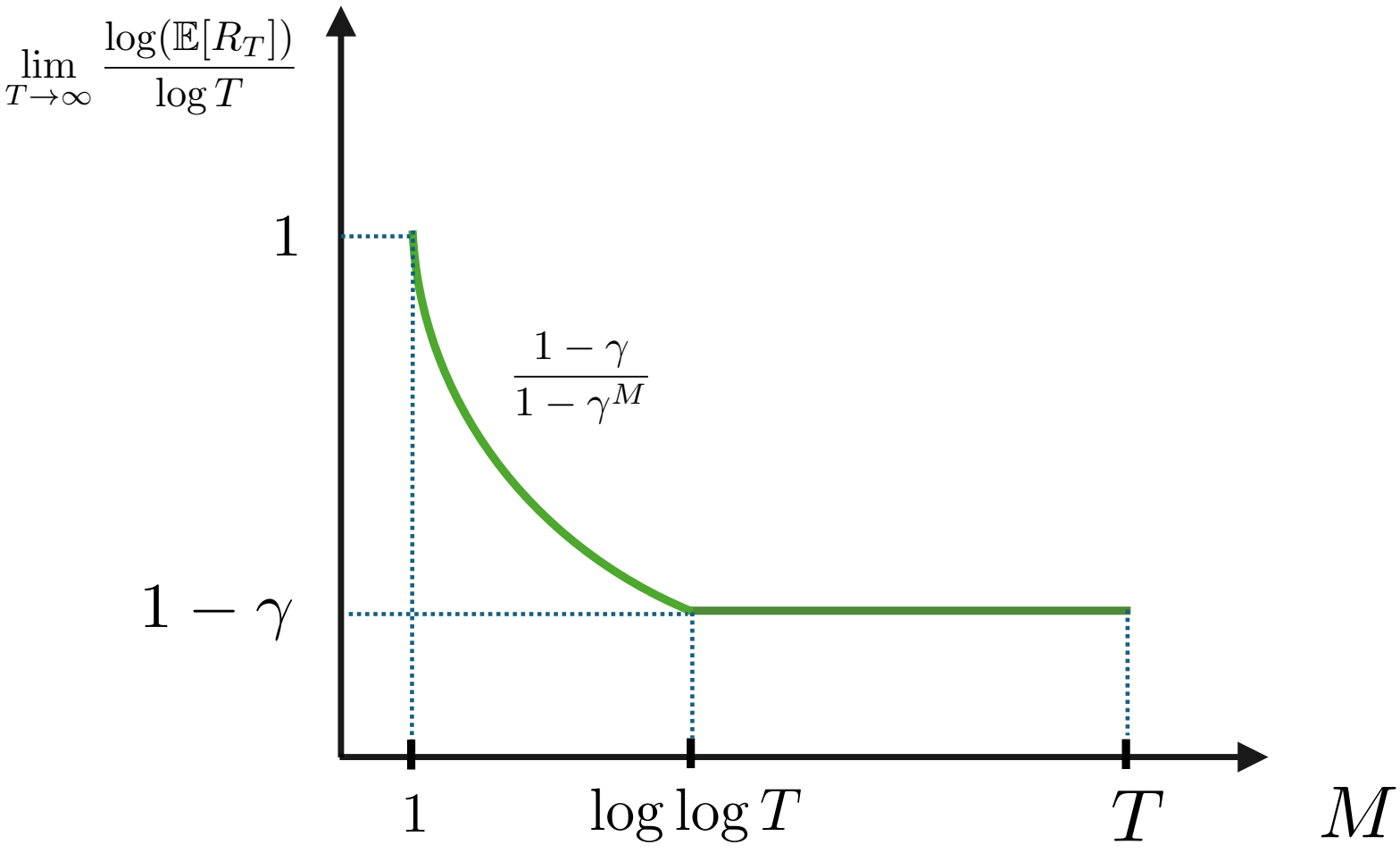
Theorem 1 (Jiang and Ma, '24)

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

$$\inf_{(\Gamma,\pi)} \sup_{P \in \mathcal{P}_{\alpha,\beta}} R_T(\Gamma, \pi; P) \asymp T^{\frac{1-\gamma}{1-\gamma^M}}$$

This holds even for adaptive grids

Our results in a figure



Key message: $\log\log T$ batches suffice to match online regret

Optimal algorithm: `BaSEDB`

Batched successive elimination with dynamic binning

Three main components in BaSEDB

- **Batch schedule:** $\Gamma = \{t_0 = 0 < t_1 < \cdots < t_M = T\}$

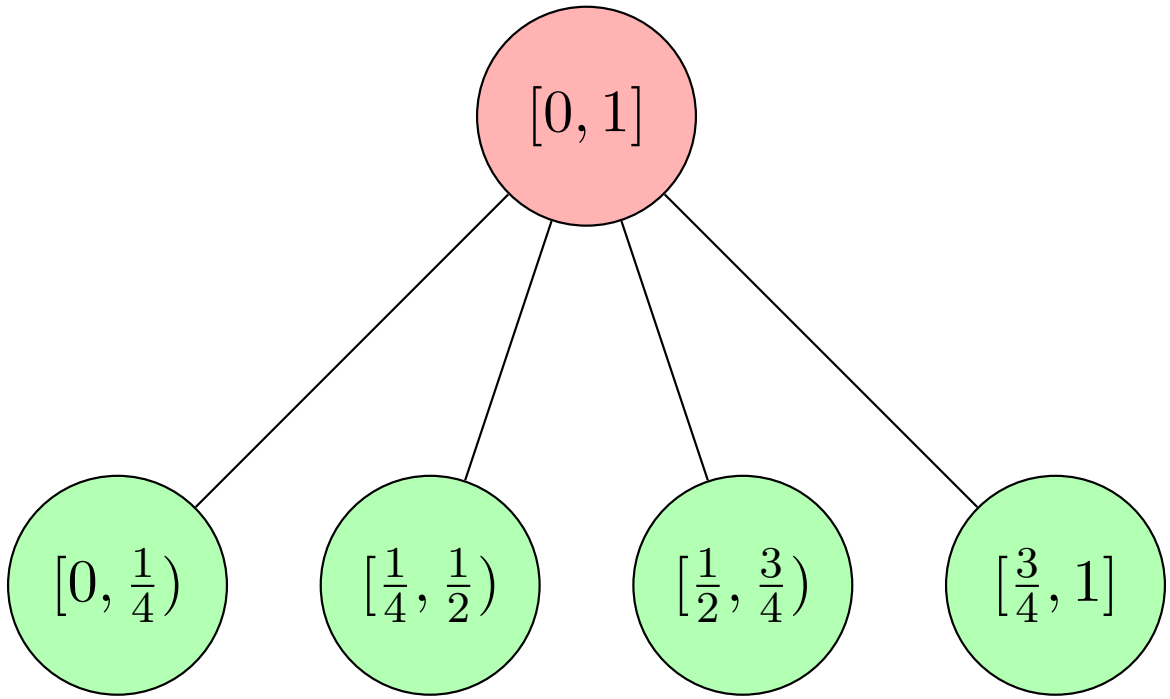
  —when to update
- **Split factors** $\{g_i\}_{i=0}^{M-1}$: control how finely we partition context space $[0, 1]^d$ into bins in each batch

  —where to focus
- **Successive elimination**: eliminate suboptimal arm in each active bin

  —what to eliminate

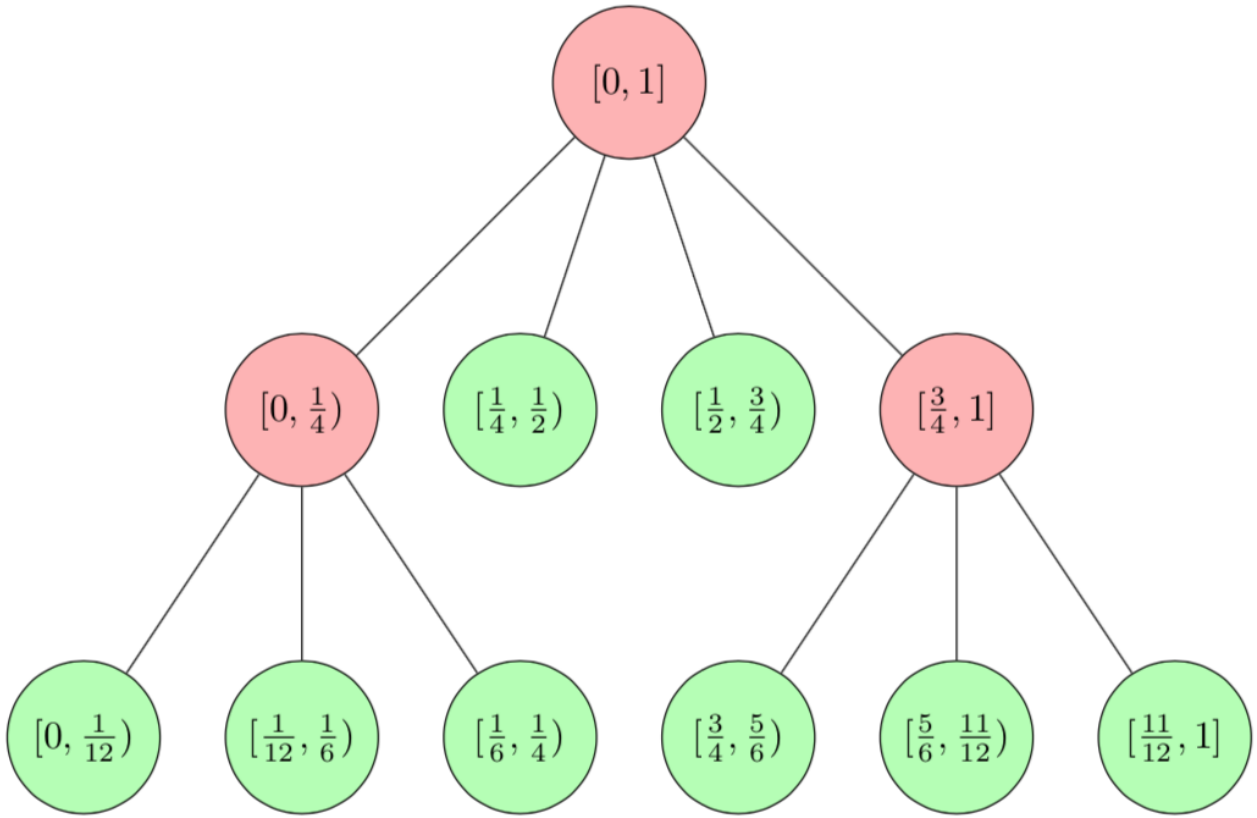
A tree diagram for BaSEDB

Prior to 1^{st} batch:



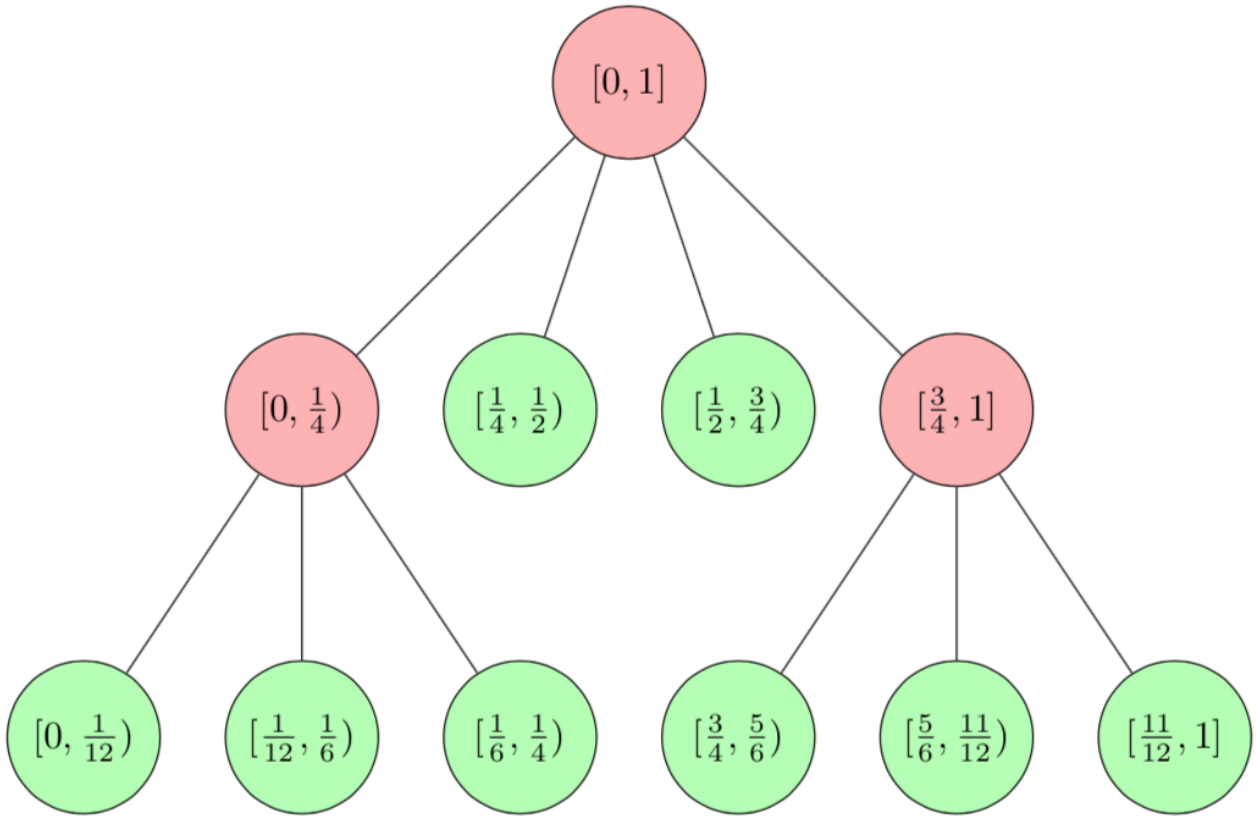
A tree diagram for BaSEDB

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



A tree diagram for BaSEDB

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



Evaluate $\rightarrow$ Eliminate $\rightarrow$ Split

Performance guarantees

- **Batch schedule:**

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

- **Split factors:**

$$g_0 = \lfloor t_1^{\frac{1}{2\beta+d}} \rfloor, \qquad \text{and} \qquad g_i = \lfloor g_{i-1}^{\gamma} \rfloor, \quad \text{for } i = 1, ..., M-2$$

Theorem 1 (Performance of `BaSEDB`)

When $M = O(\log T)$, `BaSEDB` achieves, up to log factors,

$$R_T(\widehat{\pi}) \lesssim T^{\frac{1-\gamma}{1-\gamma^M}}$$

Performance guarantees

- **Batch schedule:**

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

- **Split factors:**

$$g_0 = \lfloor t_1^{\frac{1}{2\beta+d}} \rfloor, \qquad \text{and} \qquad g_i = \lfloor g_{i-1}^{\gamma} \rfloor, \quad \text{for } i = 1, ..., M-2$$

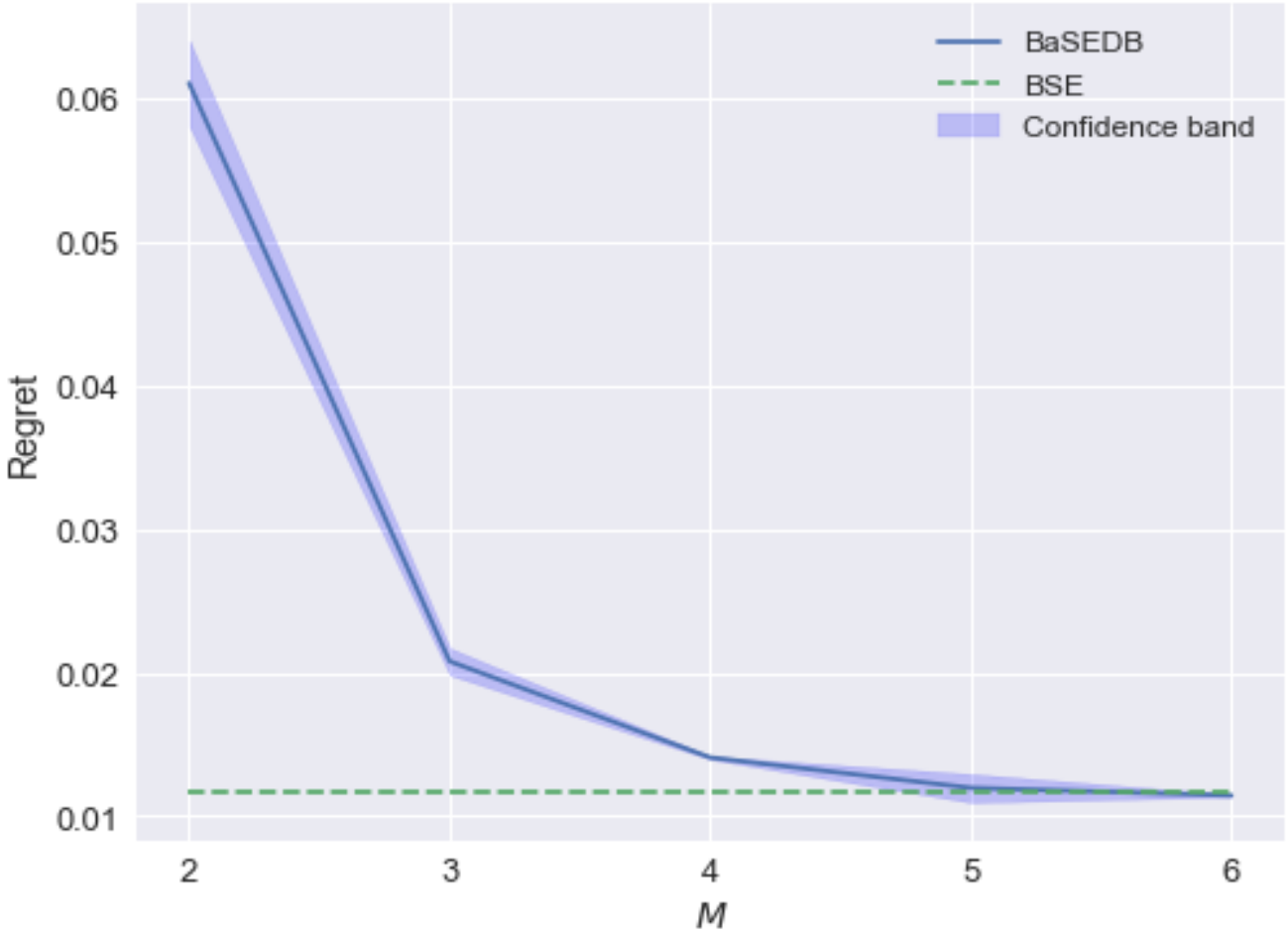
Theorem 1 (Performance of `BaSEDB`)

When $M = O(\log T)$, `BaSEDB` achieves, up to log factors,

$$R_T(\widehat{\pi}) \lesssim T^{\frac{1-\gamma}{1-\gamma^M}}$$

Feature: `BaSEDB` uses fixed grid

Numerics



$T = 50000, d = 1,\ \alpha = 0.2, \beta = 1$; BSE is optimal online policy

Part I: Takeaway

When margin is known:

- Batch constraints **do** cause performance degradation
- $\log \log T$ batches suffice to match online performance
- Limited adaptivity $\neq$ limited efficiency

What if one doesn't know the margin parameter?

Interesting observation:

- Online: adapting to unknown α is free
- Batched: `BaSEDB` requires α to set the batch schedule

What if one doesn't know the margin parameter?

Interesting observation:

- Online: adapting to unknown α is free
- Batched: `BaSEDB` requires α to set the batch schedule

Can we adapt to unknown α under batch constraints?

The challenge of unknown α

Suppose we have only two batches ($M = 2$)

- When α is **known**, we can tune the first batch length optimally:

$$t_1 \approx T^{\frac{1}{1+\gamma(\alpha)}}$$

The challenge of unknown α

Suppose we have only two batches ($M = 2$)

- When α is **known**, we can tune the first batch length optimally:

$$t_1 \approx T^{\frac{1}{1+\gamma(\alpha)}}$$

- When α is **unknown**:
  - If t_1 is too large $\Rightarrow$ waste exploration on *easy instances* (large α)

The challenge of unknown α

Suppose we have only two batches ($M = 2$)

- When α is **known**, we can tune the first batch length optimally:

$$t_1 \approx T^{\frac{1}{1+\gamma(\alpha)}}$$

- When α is **unknown**:
  - If t_1 is too large $\Rightarrow$ waste exploration on *easy instances* (large α)
  - If t_1 is too small $\Rightarrow$ underexplore and suffer on *hard instances* (small α)

The challenge of unknown α

Suppose we have only two batches ($M = 2$)

- When α is **known**, we can tune the first batch length optimally:

$$t_1 \approx T^{\frac{1}{1+\gamma(\alpha)}}$$

- When α is **unknown**:
  - If t_1 is too large $\Rightarrow$ waste exploration on *easy instances* (large α)
  - If t_1 is too small $\Rightarrow$ underexplore and suffer on *hard instances* (small α)
  - No single choice of t_1 works well for all α

The challenge of unknown α

Suppose we have only two batches ($M = 2$)

- When α is **known**, we can tune the first batch length optimally:

$$t_1 \approx T^{\frac{1}{1+\gamma(\alpha)}}$$

- When α is **unknown**:
  - If t_1 is too large $\Rightarrow$ waste exploration on *easy instances* (large α)
  - If t_1 is too small $\Rightarrow$ underexplore and suffer on *hard instances* (small α)
  - No single choice of t_1 works well for all α

This trade-off leads to the *price of unknown* α

The challenge of unknown α

Suppose we have only two batches ($M = 2$)

- When α is **known**, we can tune the first batch length optimally:

$$t_1 \approx T^{\frac{1}{1+\gamma(\alpha)}}$$

- When α is **unknown**:
  - If t_1 is too large $\Rightarrow$ waste exploration on *easy instances* (large α)
  - If t_1 is too small $\Rightarrow$ underexplore and suffer on *hard instances* (small α)
  - No single choice of t_1 works well for all α

This trade-off leads to the *price of unknown* α

Goal: Precisely characterize price of unknown margin

Regret inflation

We quantify this price by measuring how much more regret a statistician incurs compared to an oracle who knows α

Regret inflation

We quantify this price by measuring how much more regret a statistician incurs compared to an oracle who knows α

Regret inflation

$$\mathrm{RI}(\Gamma, \pi) := \sup_{\alpha \in \mathcal{K}} \sup_{P \in \mathcal{P}_\alpha} \frac{R_T(\Gamma, \pi; P)}{R_T^\star(\alpha)}$$

Interpretation:

- $\mathcal{K} := [0, d/\beta] \cup \{\infty\}$: set of possible margin parameters
- $R_T^\star(\alpha)$: best achievable regret if α were known
- RI measures how much worse we do when α is unknown

Game-theoretic view of regret inflation

- **Statistician's strategy:** M-batch policy (Γ, π) without knowledge of margin
- **Nature's strategy:** data generating process P with margin α
- **Payoff:** statistician pays ratio between her regret and the oracle's

$$\mathrm{RI}(\Gamma, \pi) = \sup_{\alpha \in \mathcal{K}} \sup_{P \in \mathcal{P}_\alpha} \frac{R_T(\Gamma, \pi; P)}{R_T^\star(\alpha)}$$

The equilibrium of this game is the "price of unknown α"

Game-theoretic view of regret inflation

- **Statistician's strategy:** M-batch policy (Γ, π) without knowledge of margin
- **Nature's strategy:** data generating process P with margin α
- **Payoff:** statistician pays ratio between her regret and the oracle's

$$\mathrm{RI}(\Gamma, \pi) = \sup_{\alpha \in \mathcal{K}} \sup_{P \in \mathcal{P}_\alpha} \frac{R_T(\Gamma, \pi; P)}{R_T^\star(\alpha)}$$

The equilibrium of this game is the "price of unknown α"

Challenge: Both players' strategy spaces are infinite-dimensional

A finite-dimensional two-player game

Let's simplify!

- **Statistician's strategy:**

$$\boldsymbol{u} \in \mathcal{U}_M = \{\boldsymbol{u} \in \mathbb{R}^{M-1} : 0 \le u_1 \le \cdots \le u_{M-1} \le 1\}$$

—think of $t_i = T^{u_i}$

- **Nature's strategy:** an $\alpha \in \mathcal{K}$
- **Payoff:** statistician pays $\Psi_M(\boldsymbol{u}, \alpha)$

A finite-dimensional two-player game

Let's simplify!

- **Statistician's strategy:**

$$\boldsymbol{u} \in \mathcal{U}_M = \{\boldsymbol{u} \in \mathbb{R}^{M-1} : 0 \le u_1 \le \cdots \le u_{M-1} \le 1\}$$

—think of $t_i = T^{u_i}$

- **Nature's strategy:** an $\alpha \in \mathcal{K}$
- **Payoff:** statistician pays $\Psi_M(\boldsymbol{u}, \alpha)$

Payoff function

$$\Psi_M(\boldsymbol{u}, \alpha) := \max_{1 \le i \le M} \eta_i(\boldsymbol{u}, \alpha) - \frac{1-\gamma(\alpha)}{1-\gamma^M(\alpha)}$$

with $\eta_1(\boldsymbol{u}, \alpha) = u_1$, $\eta_i(\boldsymbol{u}, \alpha) = u_i - u_{i-1}\gamma(\alpha)$ for $2 \le i \le M-1$, and $\eta_M(\boldsymbol{u}, \alpha) = 1 - u_{M-1}\gamma(\alpha)$

A finite-dimensional two-player game

Let's simplify!

- **Statistician's strategy:**

$$\boldsymbol{u} \in \mathcal{U}_M = \{\boldsymbol{u} \in \mathbb{R}^{M-1} : 0 \le u_1 \le \cdots \le u_{M-1} \le 1\}$$

—think of $t_i = T^{u_i}$

- **Nature's strategy:** an $\alpha \in \mathcal{K}$
- **Payoff:** statistician pays $\Psi_M(\boldsymbol{u}, \alpha)$

Feature: A simpler and solvable two-player game

Main results

Define optimal value of this finite-dimensional two-player game

$$\psi_M^\star := \inf_{\boldsymbol{u} \in \mathcal{U}_M} \sup_{\alpha \in \mathcal{K}} \Psi_M(\boldsymbol{u}, \alpha)$$

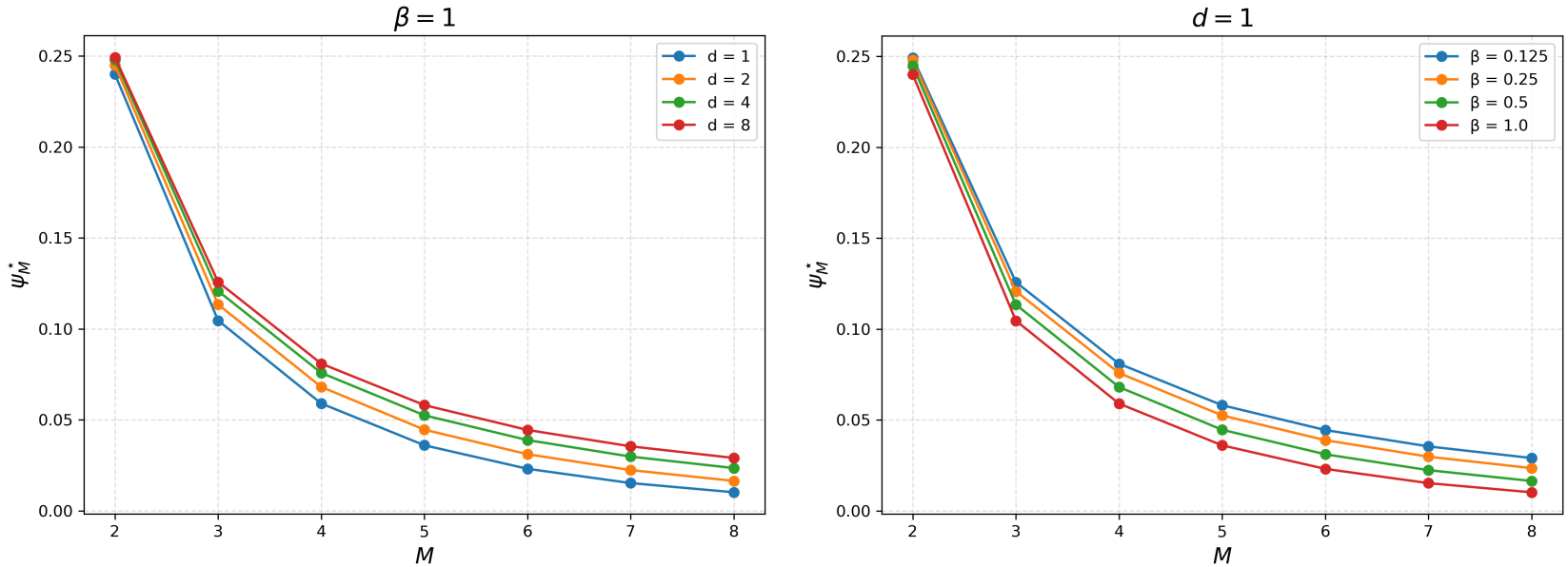
Theorem 2 (Jiang and Ma, '25)

We have, up to log factors,

$$\inf_{(\Gamma, \pi)} \mathrm{RI}(\Gamma, \pi) \asymp T^{\psi_M^\star}$$

Key message: Simpler game characterizes optimal regret inflation

Numerical illustrations



Observations:

- When M is fixed, polynomial regret inflation
- Adaptation cost is higher for higher dimension and smaller β
- As M increases, regret inflation is smaller

How many batches suffice for no regret inflation?

While we are not able to write $\psi_M^\star$ in closed-form, we have crude order-wise control:

Proposition 3

There exist $c, C \in (0,1)$ *such that*

$$c^M \leq \psi_M^\star \leq C^M$$

Consequences:

- When $M \gg \log\log T$, we have constant regret inflation
- When $M \ll \log\log T$, regret inflation is super-polylog in T

Optimal algorithm: `RoBIN`
RObust batched algorithm with adaptive BINning

Optimal algorithm: RoBIN
RObust batched algorithm with adaptive BINning

RoBIN is a robust version of BaSEDB

Recall BaSEDB

- **Batch schedule:** $\Gamma = \{t_0 = 0 < t_1 < \cdots < t_M = T\}$

  —when to update
- **Split factors** $\{g_i\}_{i=0}^{M-1}$: control how finely we partition context space into bins in each stage

  —where to focus

Both rely on knowing α to achieve optimal regret for $\mathcal{P}_\alpha$

RoBIN

Let $\boldsymbol{u}^\star$ be the minimizer of convex program

$$\inf_{\boldsymbol{u}\in\mathcal{U}_M} \sup_{\alpha\in\mathcal{K}} \Psi_M(\boldsymbol{u},\alpha)$$

- **Batch schedule:**

$$t_i \asymp T^{u_i^\star}, \quad 1 \le i \le M-1$$

- **Split factors:**

$$g_0 = \lceil T^{\frac{1}{2\beta+d}\cdot u_1^\star} \rceil, \qquad \text{and} \qquad g_i = \lceil T^{\frac{1}{2\beta+d}(u_{i+1}^\star - u_i^\star)} \rceil$$

Key: Robust parameter design informed by convex optimization

Performance guarantees for RoBIN

Theorem 4 (Performance of RoBIN)

RoBIN achieves, up to log factors,

$$\mathrm{RI}(\widehat{\Gamma}_{\textit{RoBIN}}, \widehat{\pi}_{\textit{RoBIN}}) \lesssim T^{\psi_M^\star}$$

Key messages:

- Minimax optimal regret inflation
- $\log \log T$ batches suffice for no regret inflation
- Achieves adaptivity through batch scheduling, not estimation

Part II: Takeaway

When margin parameter is unknown:

- Online: adapting to unknown α is free
- Batched: adaptation incurs polynomial regret inflation, completely characterized by $\psi_M^\star$

Closing remarks

Summary:

- Known α: batching causes performance degradation

$$\inf_{(\Gamma,\pi)} \sup_{P\in\mathcal{P}_\alpha} R_T(\Gamma,\pi;P) \asymp T^{\frac{1-\gamma(\alpha)}{1-\gamma^M(\alpha)}}$$

Closing remarks

Summary:

- Known α: batching causes performance degradation

$$\inf_{(\Gamma,\pi)} \sup_{P\in\mathcal{P}_\alpha} R_T(\Gamma,\pi;P) \asymp T^{\frac{1-\gamma(\alpha)}{1-\gamma^M(\alpha)}}$$

- Unknown α: batching causes further regret inflation

$$\inf_{(\Gamma,\pi)} \mathrm{RI}(\Gamma,\pi) \asymp T^{\psi_M^\star}$$

Closing remarks

Summary:

- Known α: batching causes performance degradation

$$\inf_{(\Gamma,\pi)} \sup_{P\in\mathcal{P}_\alpha} R_T(\Gamma,\pi;P) \asymp T^{\frac{1-\gamma(\alpha)}{1-\gamma^M(\alpha)}}$$

- Unknown α: batching causes further regret inflation

$$\inf_{(\Gamma,\pi)} \mathrm{RI}(\Gamma,\pi) \asymp T^{\psi^\star_M}$$

- $\log\log T$ batches suffice for no regret inflation and matching online performance

Closing remarks

Summary:

- Known α: batching causes performance degradation

$$\inf_{(\Gamma,\pi)} \sup_{P\in\mathcal{P}_\alpha} R_T(\Gamma,\pi;P) \asymp T^{\frac{1-\gamma(\alpha)}{1-\gamma^M(\alpha)}}$$

- Unknown α: batching causes further regret inflation

$$\inf_{(\Gamma,\pi)} \mathrm{RI}(\Gamma,\pi) \asymp T^{\psi^\star_M}$$

- $\log\log T$ batches suffice for no regret inflation and matching online performance

Moving forward:

- Joint adaptation to margin and smoothness
- What are general tools for tackling limited adaptivity?

Thank you

Papers:

- R. Jiang, and C. Ma, "Batched Nonparametric Contextual Bandits," IEEE Transactions on Information Theory, 2025.
- R. Jiang, and C. Ma, "The Adaptivity Barrier in Batched Contextual Bandits: Sharp Characterization of the Price of Unknown Margin," arxiv:2511.03708, 2025.