Batched Multi-armed Bandits Problem

Yanjun Han (Stanford EE)

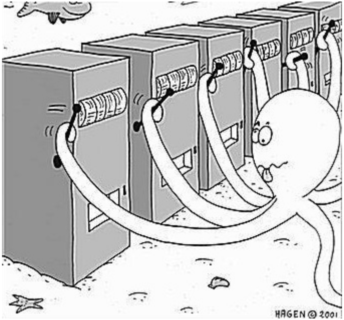
Joint work with:

Zijun Gao	Stanford Stats
Zhimei Ren	Stanford Stats
Zhengqing Zhou	Stanford Math

NeurIPS 2019, Vancouver, Canada

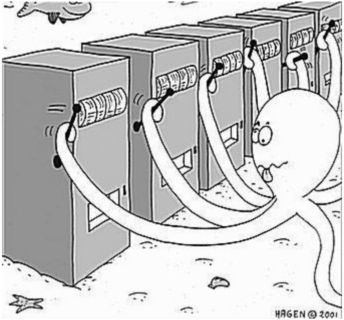
Background: Multi-armed Bandits (MAB)

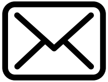
- sequential decision making
- time horizon T
- action space: K arms
- random reward for each action
- target: maximize the cumulative rewards



Background: Multi-armed Bandits (MAB)

- sequential decision making
- time horizon T
- action space: K arms
- random reward for each action
- target: maximize the cumulative rewards





Spam filtering

Dynamic pricing

Recommender system

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2									
3									
4									
5									
⋮									
K									

Partial Information in the “Space” Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2									
3	✓								
4									
5									
⋮									
K									

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2		✓							
3	✓								
4									
5									
⋮									
K									

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	…	T
1									
2		✓							
3	✓								
4									
5									
⋮									
K			✓						

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2		✓							
3	✓								
4									
5				✓					
⋮									
K			✓						

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2		✓							
3	✓				✓				
4									
5				✓					
⋮									
K			✓						

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1						✓			
2		✓							
3	✓				✓				
4									
5				✓					
⋮									
K			✓						

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1						✓			
2		✓							
3	✓				✓				
4									
5				✓					
$\vdots$									
K			✓				✓		

Partial Information in the "Space" Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	...	T
1						✓			
2		✓							
3	✓				✓				
4									
5				✓					
⋮								✓	
K			✓				✓		

Partial Information in the “Space” Domain

Space Domain: Bandit Feedback

Only the reward of the pulled arm is revealed.

Arm \ Time	1	2	3	4	5	6	7	$\dots$	T
1						✓			
2		✓							
3	✓				✓				
4									✓
5				✓					
$\vdots$								✓	
K			✓				✓		

Batched Multi-armed Bandit

Batched MAB problem:

- limited rounds of actively querying data
- split the time horizon into M batches
- rewards revealed simultaneously at the end of each batch

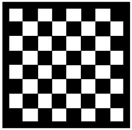
Batched Multi-armed Bandit

Batched MAB problem:

- limited rounds of actively querying data
- split the time horizon into M batches
- rewards revealed simultaneously at the end of each batch

Clinical trial

Crowdsourcing



Reinforcement learning

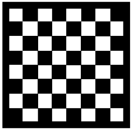
Batched Multi-armed Bandit

Batched MAB problem:

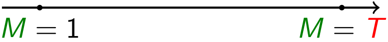
- limited rounds of actively querying data
- split the time horizon into M batches
- rewards revealed simultaneously at the end of each batch

Clinical trial

Crowdsourcing



Reinforcement learning



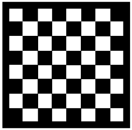
Batched Multi-armed Bandit

Batched MAB problem:

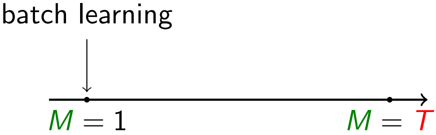
- limited rounds of actively querying data
- split the time horizon into M batches
- rewards revealed simultaneously at the end of each batch

Clinical trial

Crowdsourcing



Reinforcement learning



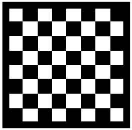
Batched Multi-armed Bandit

Batched MAB problem:

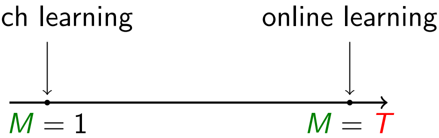
- limited rounds of actively querying data
- split the time horizon into M batches
- rewards revealed simultaneously at the end of each batch

Clinical trial

Crowdsourcing



Reinforcement learning



Partial Information in the "Time" Domain

Time Domain: Limited Rounds of Adaptivity

Feedbacks are only revealed in batches.

Partial Information in the "Time" Domain

Time Domain: Limited Rounds of Adaptivity

Feedbacks are only revealed in batches.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2									
3									
4									
5									
⋮									
K									

Partial Information in the "Time" Domain

Time Domain: Limited Rounds of Adaptivity

Feedbacks are only revealed in batches.

Arm \ Time	1	2	3	4	5	6	7	...	T
1									
2		✓							
3	✓								
4									
5									
⋮									
K			✓						

Partial Information in the "Time" Domain

Time Domain: Limited Rounds of Adaptivity

Feedbacks are only revealed in batches.

Arm \ Time	1	2	3	4	5	6	7	...	T
1						✓			
2		✓							
3	✓				✓				
4									
5				✓					
⋮									
K			✓						

Partial Information in the “Time” Domain

Time Domain: Limited Rounds of Adaptivity

Feedbacks are only revealed in batches.

Arm \ Time	1	2	3	4	5	6	7	...	T
1						✓			
2		✓							
3	✓				✓				
4									✓
5				✓					
⋮								✓	
K			✓				✓		

Mathematical Formulation

- time horizon T, number of arms K

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i \in [K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i\in[K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$
- policy π: π_t determined by the observed rewards before current batch

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i \in [K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$
- policy π: π_t determined by the observed rewards before current batch

Regret

$$R(\pi) = \sum_{t=1}^{T} \left(\mu^\star - \mu^{(\pi_t)} \right).$$

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i\in[K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$
- policy π: π_t determined by the observed rewards before current batch

Regret

$$R(\pi) = \sum_{t=1}^{T} \left(\mu^\star - \mu^{(\pi_t)} \right).$$

Batch constraint represented by a grid $t_1 < t_2 < \cdots < t_M = T$

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i\in[K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$
- policy π: π_t determined by the observed rewards before current batch

Regret

$$R(\pi) = \sum_{t=1}^{T} \left(\mu^\star - \mu^{(\pi_t)} \right).$$

Batch constraint represented by a grid $t_1 < t_2 < \cdots < t_M = T$

- static grid: $\mathcal{T} = \{t_1, \cdots, t_M\}$ fixed in advance

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i \in [K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$
- policy π: π_t determined by the observed rewards before current batch

Regret

$$R(\pi) = \sum_{t=1}^{T} \left(\mu^\star - \mu^{(\pi_t)} \right).$$

Batch constraint represented by a grid $t_1 < t_2 < \cdots < t_M = T$

- static grid: $\mathcal{T} = \{t_1, \cdots, t_M\}$ fixed in advance
- adaptive grid: the next grid point determined by historic data

Mathematical Formulation

- time horizon T, number of arms K
- stochastic MAB: pulling arm i gives reward $r_t \sim \mathcal{N}(\mu^{(i)}, 1)$
- best arm $\mu^\star = \max_{i\in[K]} \mu^{(i)}$, suboptimality gap $\Delta_i = \mu^\star - \mu^{(i)}$
- policy π: π_t determined by the observed rewards before current batch

Regret

$$R(\pi) = \sum_{t=1}^{T} \left(\mu^\star - \mu^{(\pi_t)} \right).$$

Batch constraint represented by a grid $t_1 < t_2 < \cdots < t_M = T$

- static grid: $\mathcal{T} = \{t_1, \cdots, t_M\}$ fixed in advance
- adaptive grid: the next grid point determined by historic data
- task: design policy + grid

Two Types of Regrets

Tight analysis of stochastic MAB [Vog'60, LR'85, AB'09]:

$$\mathbb{E}[R(\pi^1)] \leq C \cdot \sqrt{KT}$$

$$\mathbb{E}[R(\pi^2)] \leq C \cdot \sum_{i \neq \star} \frac{1 \vee \log(T\Delta_i^2)}{\Delta_i}$$

Two Types of Regrets

Tight analysis of stochastic MAB [Vog'60, LR'85, AB'09]:

$$\mathbb{E}[R(\pi^1)] \le C \cdot \sqrt{KT}$$

$$\mathbb{E}[R(\pi^2)] \le C \cdot \sum_{i \neq \star} \frac{1 \vee \log(T\Delta_i^2)}{\Delta_i}$$

Minimax Regret

$$R_{\text{min-max}}(K, M, T) = \inf_{\pi, \mathcal{T}} \sup_{\|\Delta\|_\infty \le \sqrt{K}} \mathbb{E}[R(\pi)]$$

Two Types of Regrets

Tight analysis of stochastic MAB [Vog'60, LR'85, AB'09]:

$$\mathbb{E}[R(\pi^1)] \leq C \cdot \sqrt{KT}$$

$$\mathbb{E}[R(\pi^2)] \leq C \cdot \sum_{i \neq \star} \frac{1 \vee \log(T\Delta_i^2)}{\Delta_i}$$

Minimax Regret

$$R_{\text{min-max}}(K, M, T) = \inf_{\pi, \mathcal{T}} \sup_{\|\Delta\|_\infty \leq \sqrt{K}} \mathbb{E}[R(\pi)]$$

Problem-dependent Regret

$$R_{\text{pro-dep}}(K, M, T) = \inf_{\pi, \mathcal{T}} \sup_{\Delta > 0} \Delta \cdot \sup_{\Delta_i \in \{0\} \cup [\Delta, \sqrt{K}]} \mathbb{E}[R(\pi)]$$

Previous Results

Full online case:

$$R_{\text{min-max}}(K, T, T) = \Theta(\sqrt{KT})$$
$$R_{\text{pro-dep}}(K, T, T) = \Theta(K \log(T))$$

Previous Results

Full online case:

$$R_{\text{min-max}}(K, T, T) = \Theta(\sqrt{KT})$$
$$R_{\text{pro-dep}}(K, T, T) = \Theta(K \log(T))$$

Required number of batches [ACBF'02, CBDS'13]:

$$R_{\text{min-max}}(K, \log T, T) = \widetilde{\Theta}(\sqrt{KT}) \quad \text{(UCB2)}$$
$$R_{\text{min-max}}(K, \log\log T, T) = \widetilde{\Theta}(\sqrt{KT}) \quad \text{(switching cost)}$$

Previous Results

Full online case:

$$R_{\text{min-max}}(K, T, T) = \Theta(\sqrt{KT})$$
$$R_{\text{pro-dep}}(K, T, T) = \Theta(K \log(T))$$

Required number of batches [ACBF'02, CBDS'13]:

$$R_{\text{min-max}}(K, \log T, T) = \widetilde{\Theta}(\sqrt{KT}) \quad \text{(UCB2)}$$
$$R_{\text{min-max}}(K, \log\log T, T) = \widetilde{\Theta}(\sqrt{KT}) \quad \text{(switching cost)}$$

Two-armed case with static grid [PRCS'16]:

$$R_{\text{min-max}}(2, M, T) = \widetilde{\Theta}(T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(2, M, T) = \widetilde{\Theta}(T^{\frac{1}{M}})$$

Previous Results

Full online case:

$$R_{\text{min-max}}(K, T, T) = \Theta(\sqrt{KT})$$
$$R_{\text{pro-dep}}(K, T, T) = \Theta(K \log(T))$$

Required number of batches [ACBF'02, CBDS'13]:

$$R_{\text{min-max}}(K, \log T, T) = \widetilde{\Theta}(\sqrt{KT}) \quad \text{(UCB2)}$$
$$R_{\text{min-max}}(K, \log\log T, T) = \widetilde{\Theta}(\sqrt{KT}) \quad \text{(switching cost)}$$

Two-armed case with static grid [PRCS'16]:

$$R_{\text{min-max}}(2, M, T) = \widetilde{\Theta}(T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(2, M, T) = \widetilde{\Theta}(T^{\frac{1}{M}})$$

Lower bounds typically very challenging [JJNZ'16, AAAK'17, DRY'18, ...].

Main Result I: Upper Bound

Theorem 1 (Upper Bound)

There exist policies π^1, π^2 such that

$$\mathbb{E}[R(\pi^1)] \le \text{polylog}(K, T) \cdot \sqrt{K} T^{\frac{1}{2-2^{1-M}}}$$

$$\mathbb{E}[R(\pi^2)] \le \text{polylog}(K, T) \cdot \frac{K T^{\frac{1}{M}}}{\min_{i \ne \star} \Delta_i}$$

Main Result I: Upper Bound

Theorem 1 (Upper Bound)

There exist policies π^1, π^2 such that

$$\mathbb{E}[R(\pi^1)] \le \text{polylog}(K, T) \cdot \sqrt{K} T^{\frac{1}{2-2^{1-M}}}$$

$$\mathbb{E}[R(\pi^2)] \le \text{polylog}(K, T) \cdot \frac{K T^{\frac{1}{M}}}{\min_{i \neq \star} \Delta_i}$$

- $M = \log\log T$ batches sufficient for centralized minimax regret
- $M = \log T$ batches sufficient for centralized problem-dependent regret

BaSE Policy

BaSE (Batched Successive Elimination)

Input: K, M, T, time grid $\mathcal{T}$

Output: policy π

BaSE Policy

BaSE (Batched Successive Elimination)

Input: K, M, T, time grid $\mathcal{T}$

Output: policy π

initialize the set of active arms $\mathcal{A} \leftarrow [K]$;

BaSE Policy

BaSE (Batched Successive Elimination)

Input: K, M, T, time grid $\mathcal{T}$

Output: policy π

initialize the set of active arms $\mathcal{A} \leftarrow [K]$;

for $m = 1$ to M **do**

pull all active arms for same number of times in m-th batch;

BaSE Policy

BaSE (Batched Successive Elimination)

Input: K, M, T, time grid $\mathcal{T}$

Output: policy π

initialize the set of active arms $\mathcal{A} \leftarrow [K]$;

for $m = 1$ to M **do**

pull all active arms for same number of times in m-th batch;

estimate the mean reward for each active arm;

BaSE Policy

BaSE (Batched Successive Elimination)

Input: K, M, T, time grid $\mathcal{T}$
Output: policy π
initialize the set of active arms $\mathcal{A} \leftarrow [K]$;
for $m = 1$ to M **do**
  pull all active arms for same number of times in m-th batch;
  estimate the mean reward for each active arm;
  eliminate all probably suboptimal arms from $\mathcal{A}$.
end for

Optimal Grid Design

Minimax Grid

$\mathcal{T}_{\text{minimax}} = \{t_1, \cdots, t_M\}$ with

$$t_1 = a, \qquad t_m = \lfloor a\sqrt{t_{m-1}} \rfloor,$$

where a is chosen such that $t_M = T$.

Optimal Grid Design

Minimax Grid

$\mathcal{T}_{\text{minimax}} = \{t_1, \cdots, t_M\}$ with

$$t_1 = a, \qquad t_m = \lfloor a\sqrt{t_{m-1}} \rfloor,$$

where a is chosen such that $t_M = T$.

Geometric Grid

$\mathcal{T}_{\text{geometric}} = \{t'_1, \cdots, t'_M\}$ with

$$t'_1 = b, \qquad t'_m = \lfloor b t'_{m-1} \rfloor,$$

where b is chosen such that $t'_M = T$.

Optimal Grid Design

Minimax Grid

$\mathcal{T}_{\text{minimax}} = \{t_1, \cdots, t_M\}$ with

$$t_1 = a, \qquad t_m = \lfloor a\sqrt{t_{m-1}} \rfloor,$$

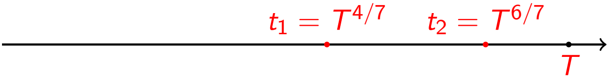
where a is chosen such that $t_M = T$.

Geometric Grid

$\mathcal{T}_{\text{geometric}} = \{t'_1, \cdots, t'_M\}$ with

$$t'_1 = b, \qquad t'_m = \lfloor b t'_{m-1} \rfloor,$$

where b is chosen such that $t'_M = T$.



Optimal Grid Design

Minimax Grid

$\mathcal{T}_{\text{minimax}} = \{t_1, \cdots, t_M\}$ with

$$t_1 = a, \qquad t_m = \lfloor a\sqrt{t_{m-1}} \rfloor,$$

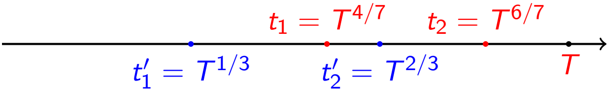
where a is chosen such that $t_M = T$.

Geometric Grid

$\mathcal{T}_{\text{geometric}} = \{t'_1, \cdots, t'_M\}$ with

$$t'_1 = b, \qquad t'_m = \lfloor b t'_{m-1} \rfloor,$$

where b is chosen such that $t'_M = T$.



Main Result II: Static Lower Bound

Theorem 2 (Static Lower Bound)

Under any static grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(\sqrt{K} T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(K, M, T) = \Omega(K T^{\frac{1}{M}})$$

Main Result II: Static Lower Bound

Theorem 2 (Static Lower Bound)

Under any static grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(\sqrt{K} T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(K, M, T) = \Omega(K T^{\frac{1}{M}})$$

- match the upper bounds within logarithmic factors

Main Result II: Static Lower Bound

Theorem 2 (Static Lower Bound)

Under any static grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(\sqrt{K} T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(K, M, T) = \Omega(K T^{\frac{1}{M}})$$

- match the upper bounds within logarithmic factors
- proof uses a max-min approach: find multiple fixed reward distributions under which no policy performs uniformly well

Max-min: Fixed Hypothesis Testing

Fundamental idea of hypothesis testing: construct several reward distributions such that

Max-min: Fixed Hypothesis Testing

Fundamental idea of hypothesis testing: construct several reward distributions such that

- Large separation: if a policy performs well under one distribution, it will perform badly under others

Max-min: Fixed Hypothesis Testing

Fundamental idea of hypothesis testing: construct several reward distributions such that

- Large separation: if a policy performs well under one distribution, it will perform badly under others
- Indistinguishability: these reward distributions are information theoretically hard to distinguish given observed rewards

Max-min: Fixed Hypothesis Testing

Fundamental idea of hypothesis testing: construct several reward distributions such that

- Large separation: if a policy performs well under one distribution, it will perform badly under others
- Indistinguishability: these reward distributions are information theoretically hard to distinguish given observed rewards

Indistinguishability Lemma

Let $Q_1, \cdots, Q_n$ be probability measures on some common probability space. Then for any tree $T = ([n], E)$ and test Ψ,

$$\frac{1}{n}\sum_{i=1}^{n} Q_i(\Psi \neq i) \geq \sum_{(i,j)\in E} \frac{1}{2n} \exp(-D_{\mathrm{KL}}(Q_i \| Q_j)).$$

Main Result III: Adaptive Lower Bound

Theorem 3 (Adaptive Lower Bound)

Under any adaptive grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(M^{-2} \cdot \sqrt{K} T^{\frac{1}{2-2^{1-M}}})$$

$$R_{\text{pro-dep}}(K, M, T) = \Omega(M^{-2} \cdot K T^{\frac{1}{M}})$$

Main Result III: Adaptive Lower Bound

Theorem 3 (Adaptive Lower Bound)

Under any adaptive grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(M^{-2} \cdot \sqrt{K} T^{\frac{1}{2-2^{1-M}}})$$

$$R_{\text{pro-dep}}(K, M, T) = \Omega(M^{-2} \cdot K T^{\frac{1}{M}})$$

- still match the upper bounds within logarithmic factors

Main Result III: Adaptive Lower Bound

Theorem 3 (Adaptive Lower Bound)

Under any adaptive grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(M^{-2} \cdot \sqrt{K}T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(K, M, T) = \Omega(M^{-2} \cdot KT^{\frac{1}{M}})$$

- still match the upper bounds within logarithmic factors
- max-min approach breaks down even for static but randomized grid

Main Result III: Adaptive Lower Bound

Theorem 3 (Adaptive Lower Bound)

Under any adaptive grid,

$$R_{\text{min-max}}(K, M, T) = \Omega(M^{-2} \cdot \sqrt{K} T^{\frac{1}{2-2^{1-M}}})$$
$$R_{\text{pro-dep}}(K, M, T) = \Omega(M^{-2} \cdot K T^{\frac{1}{M}})$$

- still match the upper bounds within logarithmic factors
- max-min approach breaks down even for static but randomized grid
- use a min-max approach instead: construct corresponding reward distributions after a policy is given

Min-max: More Details

Construct reward distributions $P_1, P_2, \cdots, P_M$ and events $A_1, \cdots, A_M$.

Min-max: More Details

Construct reward distributions $P_1, P_2, \cdots, P_M$ and events $A_1, \cdots, A_M$.

Lemma 1 (Adaptive Hypotheses)

For any policy, if $P_m(A_m)$ is not too small for some m, then the policy incurs a large regret in the worst case.

Min-max: More Details

Construct reward distributions $P_1, P_2, \cdots, P_M$ and events $A_1, \cdots, A_M$.

Lemma 1 (Adaptive Hypotheses)

For any policy, if $P_m(A_m)$ is not too small for some m, then the policy incurs a large regret in the worst case.

Lemma 2 (Covering of Events)

For any policy it holds that

$$\sum_{m=1}^{M} P_m(A_m) \geq \frac{1}{2}.$$

Concluding Remarks

Take-home message:

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design
- lower bound: a min-max approach for adaptive grids

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design
- lower bound: a min-max approach for adaptive grids

Future directions:

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design
- lower bound: a min-max approach for adaptive grids

Future directions:

- remove the M^{-2} factor in the adaptive lower bound

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design
- lower bound: a min-max approach for adaptive grids

Future directions:

- remove the M^{-2} factor in the adaptive lower bound
- generalize to adversarial and contextual bandits

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design
- lower bound: a min-max approach for adaptive grids

Future directions:

- remove the M^{-2} factor in the adaptive lower bound
- generalize to adversarial and contextual bandits
- general tools for limited rounds of adaptivity

Concluding Remarks

Take-home message:

- impact and optimal use of partial information in time domain
- upper bound: BaSE policy with optimal grid design
- lower bound: a min-max approach for adaptive grids

Future directions:

- remove the M^{-2} factor in the adaptive lower bound
- generalize to adversarial and contextual bandits
- general tools for limited rounds of adaptivity

Thank you!