Dynamic Batch Learning in High-Dimensional Sparse Linear Contextual Bandits

Zhimei Ren and Zhengyuan Zhou (2023)

Peiyao Cai March/2023

Contents

Slides partially adopted from https://zhimeir.github.io/pdf/highdim_bandit_moils.pdf

Batch Learning: Background

Linear Contextual Bandits

- Sequential Decision Problem
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector (in $\mathbb{R}^d$)
- A random reward is generated based on the chosen action
- The expectation of the reward is a linear function of the covariate
- Target: maximize the cumulative rewards

Batch Learning: Background

Linear Contextual Bandits

Clinical trial

Recommendation system

- Sequential Decision Problem
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector (in $\mathbb{R}^d$)
- A random reward is generated based on the chosen action
- The expectation of the reward is a linear function of the covariate
- Target: maximize the cumulative rewards

Batch Learning: Background

- Each action is associated with a covariate vector (in $\mathbb{R}^d$):
- At time t: observe $\{x_{t,a}\}_{a \in [K]}$
- Pick action a
- Incur reward: $r_{t,a} = x_{t,a}^T \theta^* + \xi_t$

Batch Learning: Background

- Each action is associated with a covariate vector (in $\mathbb{R}^d$):
- At time t: observe $\{x_{t,a}\}_{a\in[K]}$
- Pick action a
- Incur reward: $r_{t,a} = x_{t,a}^T\theta* + \xi_t$
- Previously, each action is associated with a parameter vector (in $\mathbb{R}^d$):
- At time t: observe x_t
- Pick action a from $\{\theta_a^*\}_{a\in[K]}$
- Incur reward $r_{t,a} = x_t^T\theta_a^* + \xi_t$

Batch Learning: Background

- Push covariate arms: Model C.
- Puch parameter arms: Model P.
- Equivalent:
- Given Model C, write $\tilde{x}_t = \{x_{t,1}^T, \ldots, x_{t,K}^T\}^T$, $\tilde{\theta}_a^* = \{0, \ldots, \theta^{*T}, \ldots, 0\}^T$.
- Given Model P, write $\tilde{x}_{t,a} = \{0, \ldots, x_t^T, \ldots, 0\}^T$, $\tilde{\theta}^* = \{\theta_1^{*T}, \ldots, \theta_K^{*T}\}^T$.

Bandit Feedback: Online Setting

The reward is immediately observed after an arm is pulled

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

Bandit Feedback: Online Setting

The reward is immediately observed after an arm is pulled

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

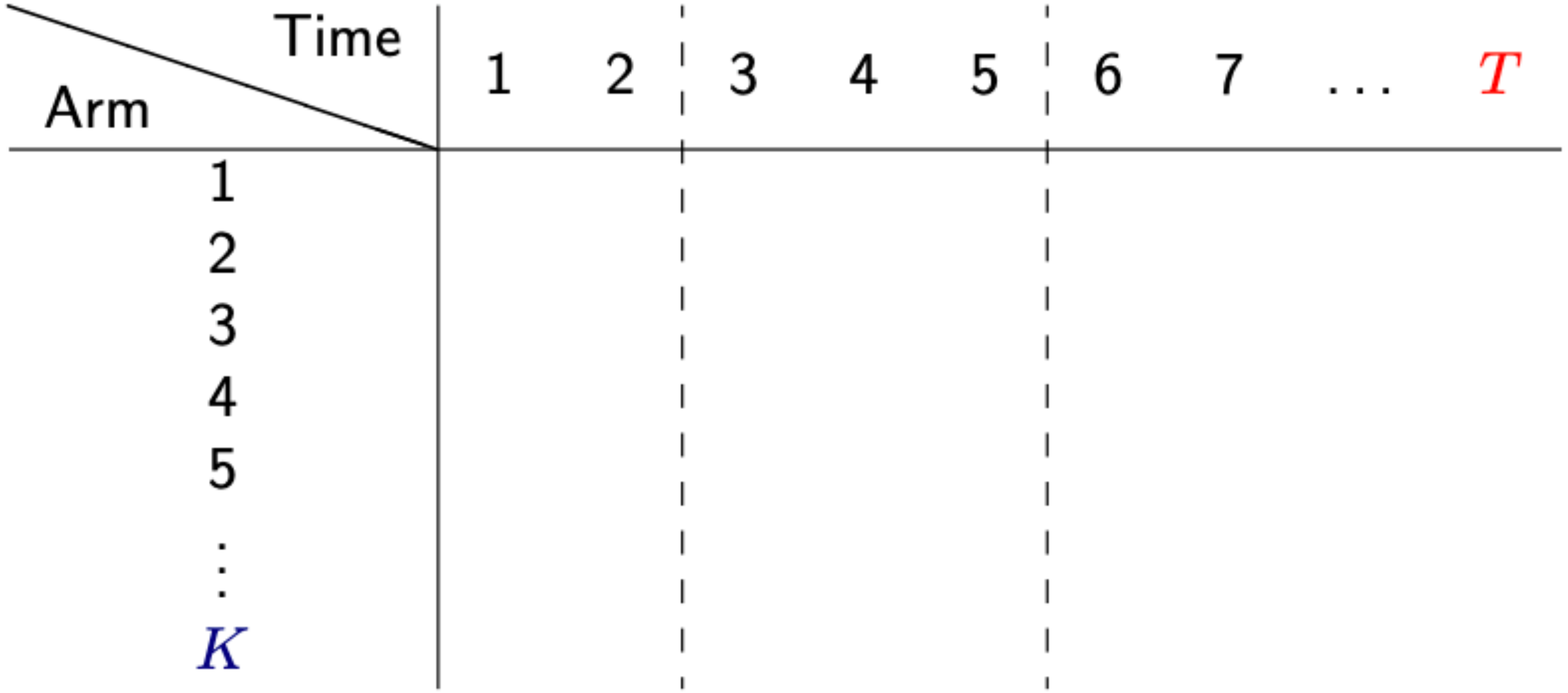
Bandit Feedback: Online Setting

The reward is immediately observed after an arm is pulled

- Online bandit learning is infeasible in practice
- Fully online learning: decision makers receive feedbacks and adjust policy immediately.
- Limited adaption due to physical cost or regulatory constrains
- Batch Constrains needed.

Bandit Feedback: Batched Case

- The time horizon is split into M batches
- The rewards can only be observed simultaneously at the end of each batch.



Bandit Feedback: Batched Case

- The time horizon is split into M batches
- The rewards can only be observed simultaneously at the end of each batch.

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

Bandit Feedback: Batched Case

- The time horizon is split into M batches
- The rewards can only be observed simultaneously at the end of each batch.

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

Bandit Feedback: Batched Case

- The time horizon is split into M batches
- The rewards can only be observed simultaneously at the end of each batch.

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

Problem Setting

This paper considers…

- Linear contexual bandits
- High-dimensional regime with sparse parameters
- Batched observations

Problem Formulation

- Time horizon T, number of arms K.
- Model C: each arm $a \in [K]$ is associated with a d-dimensional feature context $x_{t,a}$.
- The contexts $\{x_{t,a}\}_{a \in [K]}$ are iid drawn from a Kd-dimensional joint distribution.
- Select action a, incure $r_{t,a} = x_{t,a}^T \theta* + \xi_t$
- ξ_t iid zero mean sub-Gaussian RV.
- Policy $\pi = (\pi_1, \pi_t, \ldots, \pi_T)$. π_t is determined by the observed reward before the current batch.

Batch Constraint

- Number of batches: M.
- Batch constraint represented by a grid $t_1 < t_2 < \ldots < t_M = T$

- Static grid: $\mathcal{T} = \{t_1, \ldots, t_m\}$ fixed in advance.
- Adaptive grid: the next grid point determined by historic data.

- Goal: design policy + grid(?).

Metric

Regret

$$R_T(\pi, \mathcal{T}) \triangleq \sum_{t=1}^{T} \left(\max_{a \in [K]} x_{t,a}^\top \theta^\star - x_{t,a_t}^\top \theta^\star \right)$$

Minimax Regret

$$R_{\mathsf{maxmin}}(K, M, T, s_0) = \inf_{\pi, \mathcal{T}} \sup_{\|\theta^\star\|_2 \le 1, \|\theta^\star\|_0 \le s_0} \mathbb{E}\left[R_T(\pi, \mathcal{T})\right]$$

Algorithm

Lasso Batched Greedy Learning

Input Time horizon T; context dimension d; number of batches M; sparsity bound s_0.

Initialize $b = \Theta\left(\sqrt{T} \cdot \left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M-1)}}\right)$; $\hat{\theta}_0 = \mathbf{0} \in \mathbb{R}^d$;

Static grid $\mathcal{T} = \{t_1, \ldots, t_M\}$, with $t_1 = b\sqrt{s_0}$ and $t_m = b\sqrt{t_{m-1}}$ for $t \in \{2, \ldots, M\}$;

Partition each batch into M intervals evenly, i.e., $(t_{m-1}, t_m] = \cup_{j=1}^{M} T_m^{(j)}$, for $m \in [M]$.

Algorithm

Lasso Batched Greedy Learning

---

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

  for $t = t_{m-1} + 1$ to t_m **do**

    (a) Choose $a_t = \underset{a \in [K]}{\operatorname{argmax}}\ x_{t,a}^\top \hat{\theta}_{m-1}$ (break ties with lower action index).

    (b) Incur reward r_{t,a_t}.

  end for

  $T^{(m)} \leftarrow \cup_{m'=1}^{m} T_{m'}^{(m)}$; $\lambda_m \leftarrow 5\sqrt{\dfrac{2 \log K (\log d + 2 \log T)}{|T^{(m)}|}}$;

  Update $\hat{\theta}_m \leftarrow \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}}\ \dfrac{1}{2|T^{(m)}|} \sum_{t \in T^{(m)}} (r_{t,a_t} - x_{t,a_t}^\top \theta)^2 + \lambda_m \|\theta\|_1$.

end for

Grid Design

- Motivation: Sequential Batch Learning in Finite-Action Linear Contextual Bandits (Han et al 2020)
- Studied low-dimensional linear contextual batched bandit problems.
- Goal: minimizing the overall regret.
- Intuition: Optimal way of selecting grids should make sure that each batch's regret is the same (at least orderwise in terms of the dependence of T and d) - equilibrium.

Grid Design

- In Han et al, the instanteous regret at time t is at most:

$$\max_{a\in[K]}(x_{t,a} - x_{t,a_t})^\top \theta^\star \leq C'\sqrt{\log(KT)\log T} \cdot \sqrt{\frac{d}{\kappa t_{m-1}}}.$$

- t_m should cancel out the denominator $\sqrt{t_{m-1}}$ to achieve “equilibrium”.

Theory: Assumptions

Assumption 1 (sub Gaussianity)

- The marginal distribution of $x_{t,a}$ is 1-sub-Gaussian.

Theory: Assumptions

Assumption 2 (Diverse Covariate)

- There are (possibly K-dependent) positive constants $\gamma(K)$ and $\rho(K)$ such that for any $\theta \in \mathbb{R}^d$ and any unit vector $\nu \in \mathbb{R}^d$, there is $P\{(\nu^T x_{t,a*})^2 \geq \gamma(K)\} \geq \rho(K)$, where $a* = \text{argmax}_{a \in K} x_{t,a}^T \theta$.

- Intrepretation:

- Theoretical guarantee for exploration-free (greedy) algorithms.

Theory: Assumptions

Assumption 2 (Diverse Covariate)

- There are (possibly K-dependent) positive constants $\gamma(K)$ and $\rho(K)$ such that for any $\theta \in \mathbb{R}^d$ and any unit vector $\nu \in \mathbb{R}^d$, there is $P\{(\nu^T x_{t,a*})^2 \geq \gamma(K)\} \geq \rho(K)$, where $a* = \mathrm{argmax}_{a \in K} x_{t,a}^T \theta$.
- $(\nu^T x_{t,a*})^2 \rightarrow \nu^T x_{t,a*} x_{t,a*}^T \nu$
-

Theory: Assumptions

Assumption 2 (Diverse Covariate)

- There are (possibly K-dependent) positive constants $\gamma(K)$ and $\rho(K)$ such that for any $\theta \in \mathbb{R}^d$ and any unit vector $\nu \in \mathbb{R}^d$, there is $P\{(\nu^T x_{t,a*})^2 \geq \gamma(K)\} \geq \rho(K)$, where $a* = \text{argmax}_{a \in K} x_{t,a}^T \theta$.
- $(\nu^T x_{t,a*})^2 \to \nu^T x_{t,a*} x_{t,a*}^T \nu$ - RE condition in Lasso Problem.
- $\phi_{\min}(s, A) = \min_{\nu \in \mathbb{R}^d; \|\nu\|_0 \leq s} \dfrac{\nu^T A \nu}{\|\nu\|_2^2}$.
- $\phi_{\max}(s, A) = \max_{\nu \in \mathbb{R}^d; \|\nu\|_0 \leq s} \dfrac{\nu^T A \nu}{\|\nu\|_2^2}$.

Theory: Assumptions

Assumption 2 (Diverse Covariate)

- There are (possibly K-dependent) positive constants $\gamma(K)$ and $\rho(K)$ such that for any $\theta \in \mathbb{R}^d$ and any unit vector $\nu \in \mathbb{R}^d$, there is $P\{(\nu^T x_{t,a*})^2 \geq \gamma(K)\} \geq \rho(K)$, where $a* = \operatorname{argmax}_{a \in K} x_{t,a}^T \theta$.
- $(\nu^T x_{t,a*})^2 \rightarrow \nu^T x_{t,a*} x_{t,a*}^T \nu$
- RE condition leads to Lasso convergence.
-

Lemma 5. *Suppose Assumptions 1–4 hold. Given a sparsity parameter s, with probability at least* $1 - 2M^2 \exp(O(s \log d) - \Omega(\rho^2(K) \cdot \sqrt{Ts_0}/M))$, *for any* $j, m \in [M]$,

$$\phi_{\max}\left(s, \frac{D_{m,j}}{|T_m^{(j)}|}\right) \leq 16 \log K,$$

$$\phi_{\min}\left(s, \frac{D_{m,j}}{|T_m^{(j)}|}\right) \geq \frac{\gamma(K)\rho(K)}{4}.$$

Theory: Assumptions

Assumption 2 (Diverse Covariate)

- Key implication: (later we will see) the regret can be upper bounded by parameter estimation error.

- Previous concerns about the greedy algorithm: some arms may never chosen due to greedy selection, yielding inaccurate estimate of θ.

- Claim: not an issue if Hessian of the Lasso problem is well conditioned (under Assumption 2).

Theory: Assumptions

Assumption 3 (Sparsity in high-dimension)

- $d = \text{poly}(T)$
- $\|\theta\|_0 \leq s_o = O(T^{1-\epsilon})$

Theory: Assumptions

Assumption 4 (Not too many arms)

- The number of actions K satisfies $\frac{\log K}{\gamma(K)\rho(K)} = O(d/s_0)$ and $\frac{\log K}{\gamma(K)\rho(K)^3} = O(\sqrt{T^{1-\beta}/s_0})$.

Theorem 1

Regret lower bound

Theorem 1. *Fix any* s_0, d *and* T. *Let* $K = \log(T/s_0)$ *and consider the problem* $x_{t,a} \sim \mathcal{N}(0, I_d)$, $\forall a \in [K]$, $\forall t \in [T]$, *where the contexts are independence across t. Then for any* $M \leq T$ *and any dynamic batch learning algorithm* **Alg**, *we have*

$$\sup_{\theta^\star: \|\theta^\star\|_2 \leq 1, \|\theta^\star\|_0 \leq s_0} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \geq c \cdot \max\left(M^{-4} 2^{-7M/2} \cdot \sqrt{Ts_0} \cdot \left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M - 1)}}, \sqrt{Ts_0} \right), \tag{3}$$

where $\mathbb{E}_{\theta^\star}$ *denotes taking expectation w.r.t. the distribution based on the parameter* $\theta^\star$, *and* $c > 0$ *is a numerical constant independent of* (T, M, d, s_0).

Theorem 1

Regret lower bound

$$\sup_{\theta^*:\|\theta^*\|_2\le 1, \|\theta^*\|_0\le s_0} \mathbb{E}_{\theta^*}[R_T(\mathbf{Alg})] \ge c\cdot\max\left(M^{-4}2^{-7M/2}\cdot\sqrt{Ts_0}\cdot\left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M-1)}}, \sqrt{Ts_0}\right), \tag{3}$$

- The first term: depends on M.
- The second term: online regret lower bound given in “Contextual Bandits with Linear Payoff Functions” (Chu et al 2011) and Han et al 2020.
- The break-even point: $M = O\{\log\log(T/s_0)\}$.
- Smaller M: worse results, first term domination.
- Lager M: closer to online setting, second term domination.
- Good M : $O\{\log\log(T/s_0)\}$.

Theorem 1

Online Extension

Lemma 4. *When $M = T$, there exists a two-arm setting with independent Guassian contexts, for which we have (for some numerical constant c independent of T, M, d, s_0):*

$$\sup_{\theta^\star : \|\theta^\star\|_2 \le 1, \|\theta^\star\|_0 \le s_0} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \ge c \cdot \sqrt{T s_0}.$$

Theorem 2

Regret upper bound for proposed algorithm

Theorem 2. *Under* Model-C, *Assumptions* 1–4 *and* $M = O(\log\log(T/s_0))$, *we have*

$$\sup_{\theta^\star:\|\theta^\star\|_2\leq 1, \|\theta^\star\|_0\leq s_0} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \leq \frac{C \cdot M^{3/2}\sqrt{\log K \log(KT)\log(dT)}}{\gamma(K)\rho(K)} \cdot \sqrt{Ts_0}\left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M-1)}}, \tag{10}$$

where $\mathbf{Alg}$ *is LBGL and* $C > 0$ *is a numerical constant independent of* (T, d, M, K, s_0).

Theorem 2

Remarks

$$\sup_{\theta^\star:\|\theta^\star\|_2\le 1,\|\theta^\star\|_0\le s_0} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \ge c\cdot\max\left(M^{-4}2^{-7M/2}\cdot\sqrt{Ts_0}\cdot\left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M-1)}},\sqrt{Ts_0}\right),$$

$$\sup_{\theta^\star:\|\theta^\star\|_2\le 1,\|\theta^\star\|_0\le s_0} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \le \frac{C\cdot M^{3/2}\sqrt{\log K\log(KT)\log(dT)}}{\gamma(K)\rho(K)}\cdot\sqrt{Ts_0}\left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M-1)}},$$

(10

- Regret upper bound matches with the lower bound up to log factors.
- With good choice of $M = O\{\log\log(T/s_0)\}$, able to achive fully online reget $O(\sqrt{Ts_0})$ (up to log factors).

Theorem 2

Online Extension

- **Corollary 1.** *In the fully online learning setting* ($M = T$) *and under Assumptions 1–4:*

$$\sup_{\theta^\star:\|\theta^\star\|_2\leq 1, \|\theta^\star\|_0\leq s_0} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \leq \frac{C\sqrt{\left(\log\log(T/s_0)\right)^3 \log K \log(KT) \log(dT)}}{\gamma(K)\rho(K)} \cdot \sqrt{Ts_0}, \tag{11}$$

where $C > 0$ *is a numerical constant independent of* (T, d, M, K, s_0).

Conclusion

- Study the batched learning problem in high-dimensional linear contexual bandit setting.
- Develop a lower bound that characterizes the fundamental learning limits
- Provide a algorithm that yields a matching upper bound.
- Restrictions: well conditioned Hessian and knowledge about sparsity.