Dynamic Batch Learning in High-Dimensional Sparse Linear Contextual Bandits

Zhimei Ren

Informs Annual Meeting 2020

Collaborator

Zhengyuan Zhou (NYU stern)

Background: Linear Contexual Bandits

- Sequential decision making problem.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.
- Action space: K arms.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector.
- A random reward is generated based on the chosen action.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector.
- A random reward is generated based on the chosen action.
- The expectation of the reward is a linear function of the covariate chosen.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector.
- A random reward is generated based on the chosen action.
- The expectation of the reward is a linear function of the covariate chosen.
- Target: maximize the cumulative rewards.

Background: Linear Contexual Bandits

- Sequential decision making problem.
- Time horizon: T.
- Action space: K arms.
- Each action is associated with a covariate vector.
- A random reward is generated based on the chosen action.
- The expectation of the reward is a linear function of the covariate chosen.
- Target: maximize the cumulative rewards.

Clinical trial

Recommendation system

Bandit feedback: online case

- The reward is immediately observed after an arm is pulled.

Bandit feedback: online case

- 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									
$\vdots$									
K									

Bandit feedback: online case

- 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									
$\vdots$									
K									

Bandit feedback: online case

- 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 case

- 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									
$\vdots$									
K			✓						

Bandit feedback: online case

- 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 case

- 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 case

- 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 case

- 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				✓					
$\vdots$									
K			✓				✓		

Bandit feedback: online case

- 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				✓					
$\vdots$								✓	
K			✓				✓		

Bandit feedback: online case

- 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				✓					
$\vdots$								✓	
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.

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									
$\vdots$									
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	$\dots$	T
1									
2		✓							
3	✓								
4									
5									
$\vdots$									
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

We study the sequential decision making problem in

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

Problem Setting

We study the sequential decision making problem in

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

Clinical trial

Recommendation system

Mathematical Formulation

- Time horizon T; (finite) number of arms K;

Mathematical Formulation

- Time horizon T; (finite) number of arms K;
- Each arm is associated with a d-dimensional feature context $x_{t,a}$;

Mathematical Formulation

- Time horizon T; (finite) number of arms K;
- Each arm is associated with a d-dimensional feature context $x_{t,a}$;
- If a decision maker selects action $a \in [K]$, a reward $r_{t,a} \in \mathbb{R}$ is incurred:

$$r_{t,a} = x_{t,a}^\top \theta^\star + \xi_t.$$

Mathematical Formulation

- Time horizon T; (finite) number of arms K;
- Each arm is associated with a d-dimensional feature context $x_{t,a}$;
- If a decision maker selects action $a \in [K]$, a reward $r_{t,a} \in \mathbb{R}$ is incurred:

$$r_{t,a} = x_{t,a}^\top \theta^\star + \xi_t.$$

- $\theta^\star \in \mathbb{R}^d$ is the underlying unknown parameter vector; $\{\xi_t\}_{t \geq 1}$ is a sequence of i.i.d. zero-mean 1-sub-Gaussian random variables.

Mathematical Formulation

- Time horizon T; (finite) number of arms K;
- Each arm is associated with a d-dimensional feature context $x_{t,a}$;
- If a decision maker selects action $a \in [K]$, a reward $r_{t,a} \in \mathbb{R}$ is incurred:

$$r_{t,a} = x_{t,a}^\top \theta^\star + \xi_t.$$

- $\theta^\star \in \mathbb{R}^d$ is the underlying unknown parameter vector; $\{\xi_t\}_{t \geq 1}$ is a sequence of i.i.d. zero-mean 1-sub-Gaussian random variables.
- Policy π: π_t is determined by the observed rewards before the current batch.

Mathematical Formulation

- Time horizon T; (finite) number of arms K;
- Each arm is associated with a d-dimensional feature context $x_{t,a}$;
- If a decision maker selects action $a \in [K]$, a reward $r_{t,a} \in \mathbb{R}$ is incurred:

$$r_{t,a} = x_{t,a}^\top \theta^\star + \xi_t.$$

- $\theta^\star \in \mathbb{R}^d$ is the underlying unknown parameter vector; $\{\xi_t\}_{t\geq 1}$ is a sequence of i.i.d. zero-mean 1-sub-Gaussian random variables.
- Policy π: π_t is determined by the observed rewards before the current batch.

Regret:

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

Mathematical Formulation

- Time horizon T; (finite) number of arms K;
- Each arm is associated with a d-dimensional feature context $x_{t,a}$;
- If a decision maker selects action $a \in [K]$, a reward $r_{t,a} \in \mathbb{R}$ is incurred:

$$r_{t,a} = x_{t,a}^\top \theta^\star + \xi_t.$$

- $\theta^\star \in \mathbb{R}^d$ is the underlying unknown parameter vector; $\{\xi_t\}_{t \geq 1}$ is a sequence of i.i.d. zero-mean 1-sub-Gaussian random variables.
- Policy π: π_t is determined by the observed rewards before the current batch.

Regret:

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

Mathmatical Formulation

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

Mathmatical Formulation

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

Mathmatical Formulation

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

Mathmatical Formulation

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

Minimax Regret:

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

Assumptions

- $\|\theta^\star\|_2 \leq 1$.

Assumptions

- $\|\theta^\star\|_2 \leq 1$.
- $\{x_{t,a}\}_{a\in[K]}$ i.i.d. drawn from a Kd-dimensional joint distribution.

Assumptions

- $\|\theta^\star\|_2 \leq 1$.
- $\{x_{t,a}\}_{a\in[K]}$ i.i.d. drawn from a Kd-dimensional joint distribution.
- Sparsity: $d = \mathsf{Poly}(T)$; $\|\theta^\star\|_0 \leq s_0 = O(T^{1-\varepsilon})$.

Assumptions

- $\|\theta^\star\|_2 \leq 1$.
- $\{x_{t,a}\}_{a\in[K]}$ i.i.d. drawn from a Kd-dimensional joint distribution.
- Sparsity: $d = \mathsf{Poly}(T)$; $\|\theta^\star\|_0 \leq s_0 = O(T^{1-\varepsilon})$.
- Sub-Gaussianity: $x_{t,a}$ is 1-sub-Gaussian marginally.

Assumptions

- $\|\theta^\star\|_2 \le 1$.
- $\{x_{t,a}\}_{a\in[K]}$ i.i.d. drawn from a Kd-dimensional joint distribution.
- Sparsity: $d = \mathrm{Poly}(T)$; $\|\theta^\star\|_0 \le s_0 = O(T^{1-\varepsilon})$.
- Sub-Gaussianity: $x_{t,a}$ is 1-sub-Gaussian marginally.
- Restricted Bounded Density: There exists a constant $\gamma > 0$, such that for each $a \in [K]$, any subset $S \subset [d]$ with $|S| = s_0$, and any unit vector $v \in R^{s_0}$, the probability density function of $v\top x_{t,a}(S)$ exists and is bounded above by $\gamma/2$.

Assumptions

- $\|\theta^\star\|_2 \leq 1$.
- $\{x_{t,a}\}_{a\in[K]}$ i.i.d. drawn from a Kd-dimensional joint distribution.
- Sparsity: $d = \text{Poly}(T)$; $\|\theta^\star\|_0 \leq s_0 = O(T^{1-\varepsilon})$.
- Sub-Gaussianity: $x_{t,a}$ is 1-sub-Gaussian marginally.
- Restricted Bounded Density: There exists a constant $\gamma > 0$, such that for each $a \in [K]$, any subset $S \subset [d]$ with $|S| = s_0$, and any unit vector $v \in R^{s_0}$, the probability density function of $v\top x_{t,a}(S)$ exists and is bounded above by $\gamma/2$.
- Not too many arms: $K^2 \log K = O(d/s_0)$.

Previous results

Two-arm batched bandits with static grids [PRCS'16]:

$$R_{\mathsf{maxmin}}(2, M, T, 1) = \tilde{\Theta}(T^{\frac{1}{2-2^{1-M}}})$$

Multi-arm batched bandits with adaptive grids [GHRZ'19]

$$R_{\mathsf{maxmin}}(K, M, T, 1) = \tilde{\Theta}(\sqrt{K}T^{\frac{1}{2-2^{1-M}}})$$

Batched contexual bandits in low dimensions [HZZBGY'20]

$$R_{\mathsf{maxmin}}(M, T, d) = \tilde{\Theta}\left(\sqrt{dT}\left(T/d^2\right)^{\frac{1}{2(2^M-1)}}\right)$$

Online contexual bandits in high dimensions (with margin conditions) [BB'20]

$$R_{\mathsf{maxmin}}(T, T, s_0) = O\left(s_0^2(\log d + \log T)^2\right)$$

[WWY'18]

$$R_{\mathsf{maxmin}}(T, T, s_0) = O\left(s_0^2(\log d + s_0)\log T\right)$$

Our results

- High-dimensional batched contexual bandits.
- Adaptive grid design.

Our results

- High-dimensional batched contexual bandits.
- Adaptive grid design.

When $M = O(\log\log(T/s_0))$

$$R_{\mathsf{maxmin}}(M, T, s_0) = \tilde{\Theta}\left(\sqrt{Ts_0}\,(T/s_0)^{\frac{1}{2(2^M-1)}}\right)$$

Our results

- High-dimensional batched contexual bandits.
- Adaptive grid design.

When $M = O(\log\log(T/s_0))$

$$R_{\mathsf{maxmin}}(M, T, s_0) = \tilde{\Theta}\left(\sqrt{Ts_0}\,(T/s_0)^{\frac{1}{2(2^M-1)}}\right)$$

Fully online

$$R_{\mathsf{maxmin}}(T, T, s_0) = \tilde{\Theta}(\sqrt{Ts_0})$$

Lower Bound

Consider the two-action setting where $x_{t,1} \stackrel{iid}{\sim} \mathcal{N}(0, I_d)$, $x_{t,2} \stackrel{iid}{\sim} \mathcal{N}(0, I_d)$ and $x_{t,1}$ is independent of $x_{t,2}$. Then for any $M \leq T$ and any dynamic batch learning algorithm $\mathbf{Alg}$, we have:

$$\sup_{\theta^\star} \mathbb{E}_{\theta^\star}[R_T(\mathbf{Alg})] \geq c \cdot \max\left(M^{-2}\sqrt{Ts_0}\left(\frac{T}{s_0}\right)^{\frac{1}{2(2^M-1)}}, \sqrt{Ts_0} \right)$$

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

Proof Sketch

- Lower bound the worst-case regret by a sequence of Bayesian regrets $\{Q_m\}_{m\in[m]}$, each of which corresponds to a particular prior on $\theta^\star$.

Proof Sketch

- Lower bound the worst-case regret by a sequence of Bayesian regrets $\{Q_m\}_{m \in [m]}$, each of which corresponds to a particular prior on $\theta^\star$.
- Given an **Alg**, we now define for each $m \in [M]$ the "bad" event $A_m = \{t_{m-1} \le T_{m-1} < T_m \le t_m\}$ (why?)

Proof Sketch

- Lower bound the worst-case regret by a sequence of Bayesian regrets $\{Q_m\}_{m\in[m]}$, each of which corresponds to a particular prior on $\theta^\star$.
- Given an **Alg**, we now define for each $m \in [M]$ the "bad" event $A_m = \{t_{m-1} \leq T_{m-1} < T_m \leq t_m\}$ (why?)
- Show that at least one A_m occurs with a large enough probability under the corresponding prior.

Upper Bound

Under the assumptions and $M = O(\log\log(T/s_0))$, we have

$$\sup_{\theta^\star : \|\theta^\star\|_2 \le 1, \|\theta^\star\|_0 \le s_0} \mathbb{E}_{\theta^\star}[R_T(\mathsf{Alg})] = \tilde{O}\left(\sqrt{Ts_0}(T/s_0)^{\frac{1}{2(2^M-1)}}\right)$$

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

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

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

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

end for

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.

Dynamic Batch Learning in High-Dimensional Sparse linear Contextual Bandits

(https://arxiv.org/abs/2008.11918)