Multi-player Multi-armed Bandits with Delayed Feedback

Jingqi Fan[1], Zilong Wang[2], Shuai Li[2], Linghe Kong[2]

1 Northeastern University, China 2 Shanghai Jiao Tong University

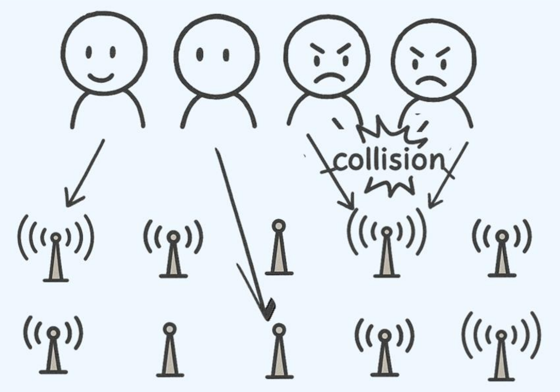
IJCAI International Joint Conferences on Artificial Intelligence Organization

Paper ID: 1412

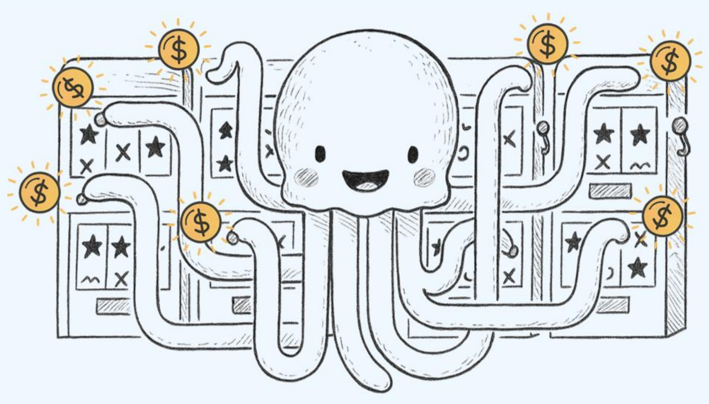
Abstract

Motivation:

Users experience delay in cognitive radio networks.



Multi-armed Bandits is a classical decision-making framework.



Contribution:

- **A new framework:** Decentralized Multi-player multi-armed bandits with stochastic delay feedback.
- **An noval algortihm:** (1) Collision-free exploration: Design specific arm-selection strategies for players to avoid collisions during exploration. (2) Implicit communication: Enable players to leverage the exploration results of others.
- **Near-optimal regret bound:** We establish a regret upper bound and derive a correspondinglower bound to prove the algorithm is near-optimal.

Setting

Problem Formulation:

- M players, K arms, T total steps.
- Let $[M] := \{1, ..., M\}$ and $[K] := \{1, ..., K\}$.
- At each step s, each player $j \in [M]$ pulls an arm $\pi^j(t) \in [K]$.
- The environment generates $X^j(s) \sim \mathrm{Bernoulli}(\mu_{\pi^j(s)})$ and $r^j(s) := X^j(s)[1 - \eta^j(s)]$.
- The environment also generates $d^j(s) \sim D_{\pi^j(s)}$, where $D_{\pi^j(s)}$ is an unknow distribution.
- Then, at step $s + d^j(s) - 1$, player j receives the feedback $[r^j(s), \eta^j(s), s]$.

Goal: minimize the regret

$$\mathbb{E}[R(T)] := \sum_{s\le T}\sum_{k\le M}\mu_k - \mathbb{E}\left[\sum_{s\le T}\sum_{j\le M} r^j(s)\right],$$

where μ_k is the k-th biggest reward expectation. $\mu_1 \ge \cdots \ge \mu_K$.

Assumption:

1. $D_k = D_{k'} = D, \forall k \in [K]$. D is sub-Gaussian.
   - σ_d^2 denotes the sub-Gaussian parameter and $\mathbb{E}[d]$ denotes the expectation.
   - Note that σ_d^2 and $\mathbb{E}[d]$ are unknown.
2. Each player is aware of her own rank j.

Algorithm

Definition:

- $N_k^j(t) := \sum_{s\le t}\mathbb{1}[\pi^j(s) = k, \eta_k(s) = 0]$ denotes the number of accumulated time steps that player j pulls arm k without collisions.
- $n_k^j(t) := \sum_{s\le t}\mathbb{1}[\pi^j(s) = k, \eta_k(s) = 0, d^j(s) + s \le t]$ denotes the number of accumulated time steps that player j pulls arm k and receive the feedback without collisions.
- Let $\mathcal{M}^j(p)$ denote the set of empirical optimal arms during the p-th phase. $|\mathcal{M}^j(p)| = M$.
- The estimated reward expectation of arm k from player j's view at step t is defined as

$$\hat{\mu}_k^j(t) := \frac{\sum_{s\le t} r^j(s)\mathbb{1}[\pi^j(s) = k, \eta_k(s) = 0, d^j(s) + s \le t]}{n_k^j(t)}.$$

- The upper confidence bound and lower confidence bound are defined as

$$\mathrm{UCB}_k^j(t) := \hat{\mu}_k^j(t) + \sqrt{\frac{2\log T}{n_k^j(t)}}, \quad \mathrm{LCB}_k^j(t) := \hat{\mu}_k^j(t) - \sqrt{\frac{2\log T}{n_k^j(t)}}.$$

Brief Introduction of the Algorithm:

- The algorithm is divided into many exploration-communication phases.
- Let $\mathcal{M}^j(p)$ denote the set of empirical optimal arms during the p-th phase. $|\mathcal{M}^j(p)| = M$. Players initialize $\mathcal{M}^j(1)$ which is a list with $\mathcal{M}^j(1) = \mathcal{M}^{j'}(1)$ for any $j, j' \in [M]$.
- Players are divided into a leader and many followers.
- They pull arms in a round-robin way to avoid collisions while the leader is in charge of exploring arms. **[Exploration Phase]**
- Sometimes they collide on purpose to pass messages. **[Communication Phase]**
- When a player j receives a feedback at time t, she updates $n_k^j(t), \hat{\mu}_k^j(t), \mathrm{UCB}_k^j(t), \mathrm{LCB}_k^j(t)$ and the estimation of $\mathbb{E}[d]$ and σ_d^2 with

$$\hat{\mu}_d^j(t) := \frac{\sum_{s<t}\left(d^j(s)\mathbb{1}\{s + d^j(s) < t\}\right)}{\sum_{s<t}\mathbb{1}\{s + d^j(s) < t\}},$$

$$(\hat{\sigma}_d^2)^j(t) := \frac{\sum_{s<t}\left([d^j(s) - \hat{\mu}_d^j(t)]\mathbb{1}\{s + d^j(s) < t\}\right)^2}{\sum_{s<t}\mathbb{1}\{s + d^j(s) < t\}}.$$

Applying 26Fall PhD

Jingqi Fan is applying **26Fall PhD**. Feel free to reach out!

- Junior **undergrad** at NEU, China.
- Research interests: **RL** (theory + real-world applications), with a specific focus on **multi-agent systems**.
- Some experience on bandits, LLM agents and RL on OM.
- Fortunate to have worked with many nice profs and interned at Theory Center & ML Group @ **MSR Asia**.

WeChat LinkedIn

- Each player j aims to find q^j such that

$$q^j = \arg\min_q \left\{ q \in \mathbb{N} \mid t > \hat{\mu}_d^j(t) + (p - q)KM\log(T)\sqrt{2(\hat{\sigma}_d^2)^j(t)\log\left((M-1)(K+2M)(T)\right)} \right\}.$$

- Starting from Phase 2, players always use some old exploration results, i.e., $\mathcal{M}^j(p - q^j)$, to mitigate the influence of delay.

Exploration Phase p

Leader:

- Explore arms in $\mathcal{M}^j(p-q^j)$ and $\mathcal{K} \setminus \mathcal{M}^j(p-q^j)$.
- Add arms with the first M-th higgest reward means into $\mathcal{M}^j(p+1)$.
- Remove arm k from $\mathcal{K}$ if $\mathrm{UCB}_k^j(t) \le \mathrm{LCB}_\ell^j(t), \forall \ell \in \mathcal{K} \setminus \{k\}$.

Followers:

- Explore arms in $\mathcal{M}^j(p-q^j)$

Communication Phase p

- When $t = p \cdot KM\log T$, a Com phase starts.
- Compare $\mathcal{M}^j(p-q^j)$ with $\mathcal{M}^j(p+1)$.
- Send i_{k^-} by pulling the i_{k^-}-th arm in $\mathcal{M}^j(p-q^j)$ for M steps.
- Send k^+ by pulling arm k^+ for K steps.
- Send $\mathrm{End} = \mathtt{False}$ by pulling arms in $\mathcal{M}^j(p-q^j)$ round-robinly for M steps.
- Receive a collision from the i_{k^-}-th arm by pulling arms in $\mathcal{M}^j(p-q^j)$ round-robinly.
- Receive a collision from arm k by pulling arms in $[K]$ round-robinly.
- Receive non-collision (indicating $\mathrm{End} = \mathtt{False}$) when pulling arms in $\mathcal{M}^j(p-q^j)$ round-robinly.

.....

Exploitation Phase

- Each player j pulls the j-th arm in $\mathcal{M}^j(p_{\max})$ until T.

Analysis

Theorem 1 [Regret Upper Bound in Decentralized setting]. Let $\Delta := \min_{k\le M}\mu_k - \mu_{k+1}$ and $\Delta_{k,\ell} := \mu_k - \mu_\ell$. In decentralized setting, given any K, M and a quantile $\theta \in (0,1)$, for delay distribution under Assumption 1, the regret of the algorithm satisfies

$$\mathbb{E}[R(T)] \le \sum_{k>M}\frac{323\log(T)}{\theta\Delta_{M,k}} + \frac{M}{K-M}\sum_{k>M}\Delta_{1,k}d_1 + \frac{15}{\theta}d_2 + d_3 + C.$$

Corollary 1 [Regret Upper Bound in centralized setting]. In centralized setting, for delay distribution under Assumption 1, given any K, M, μ and a quantile $\theta \in (0,1)$, the regret of our algorithm satisfies

$$\mathbb{E}[R(T)] \le \sum_{k>M}\frac{323\log(T)}{\theta\Delta_{M,k}} + \frac{M}{K-M}\sum_{k>M}\Delta_{1,k}\mathbb{E}[d] + \frac{9}{\theta}d_2 + d_3.$$

Theorem 2 [Regret Lower Bound]. For a quantile $\theta \in (0,1)$ and any sub-optimal gap set $S_\Delta = \{\Delta_{M,k} \mid \Delta_{M,k} = \mu_{(M)} - \mu_{(k)} \in [0,1]\}$ of cardinality $K - M$, there exists an instance with an order on S_Δ and a sub-Gaussian delay distribution such that

$$\mathbb{E}[R(T)] \ge \sum_{k>M}\frac{(1-o(1))\log(T)}{2\theta\Delta_{M,k}} + \frac{M}{2K}\sum_{k>M}\Delta_{M,k}d_4 - \frac{2}{\theta},$$

where

$$d_1 = 2\mathbb{E}[d] + \sigma_d\sqrt{3\log(K)}, \quad d_2 = \mathbb{E}[d] + \sigma_d\sqrt{2\log\left(\frac{1}{1-\theta}\right)},$$

$$d_3 = \frac{656\sqrt{2\sigma_d^2}}{\theta K^2M^2} + 3\sqrt{6}\sigma_d, \quad d_4 = \mathbb{E}[d] - \sigma_d\sqrt{\frac{\theta}{1-\theta}}, C = \sum_{k>M}\frac{195}{\theta\Delta_{M,k}} + \frac{4M}{\Delta^2}.$$

- Only the first terms in Theorem 1 and Theorem 2 are related to T. They are aligned up to constant factors.
- The last term in Theorem 1 arises from the decentralized environment and is not related to delay. In the centralized setting, this term vanishes, as shown in Corollary 1.
- Regarding delay, a comparison between Theorem 1 and Theorem 2 reveals that the difference in their dependence on K and M is only $O(\frac{1}{1-M/K})\sqrt{\log(K)}$. This indicates that the regret caused by delay does not grow rapidly as K and M increase.
- Theorem 1 and Corollary 1 demonstrate that our algorithm works efficiently in both centralized and decentralized settings.