The Hardness Analysis of Thompson Sampling for Combinatorial Semi-bandits with Greedy Oracle

Fang Kong[1], Yueran Yang[1], Wei Chen[2], Shuai Li[1]

[1]John Hopcroft Center for Computer Science, Shanghai Jiao Tong University

[2]Microsoft Research Asia

NeurIPS 2021

Stochastic multi-armed bandit

- T-round learning game between the player and m arms
- each arm $i \in [m]$ is associated with an unknown fixed reward distribution D_i with unknown mean μ_i

Stochastic multi-armed bandit

Time	1	2	3	4	5	6	7	8	9	10	11	12
Left arm	$1	$0			$1	$1	$0					
Right arm			$1	$0								

- for round $t = 1, 2, \ldots$:
- the player selects an arm $A_t \in [m]$
- then receives a reward $X_{t,A_t} \sim D_{A_t}$
- The goal of the player is to maximize the cumulative expected reward, or equivalent to minimizing the cumulative expected regret

$$\mathbb{E}\left[T \cdot \mu^* - \sum_{t=1}^{T} X_{t,A_t}\right] \tag{1}$$

Stochastic multi-armed bandit

To achieve this long-horizon goal:

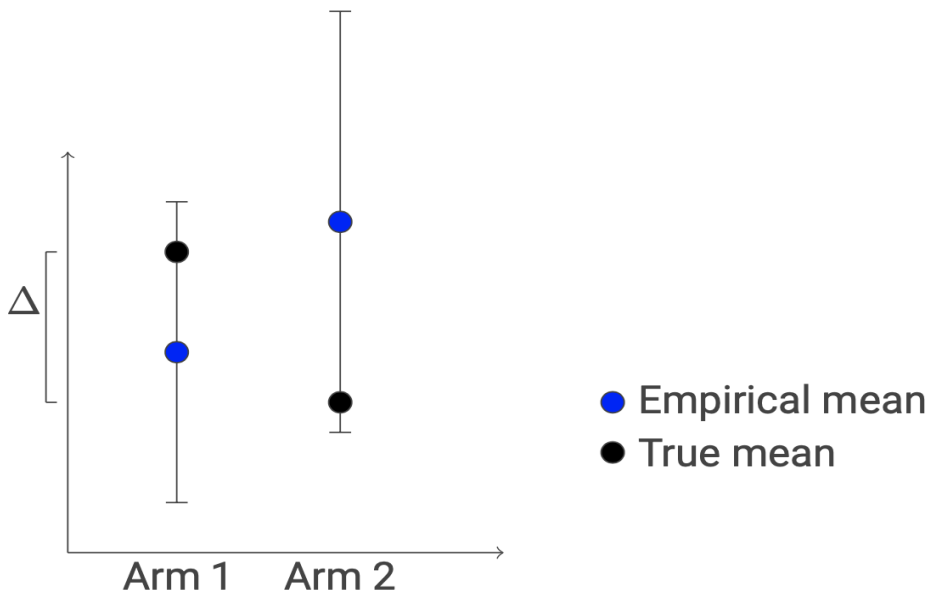
Exploration V.S. Exploitation

- exploration: try arms that have not been observed enough times
- exploitation: focus on the arm with the best observed performance so far

How to balance exploration and exploitation is the key of algorithms

upper confidence bound (UCB)

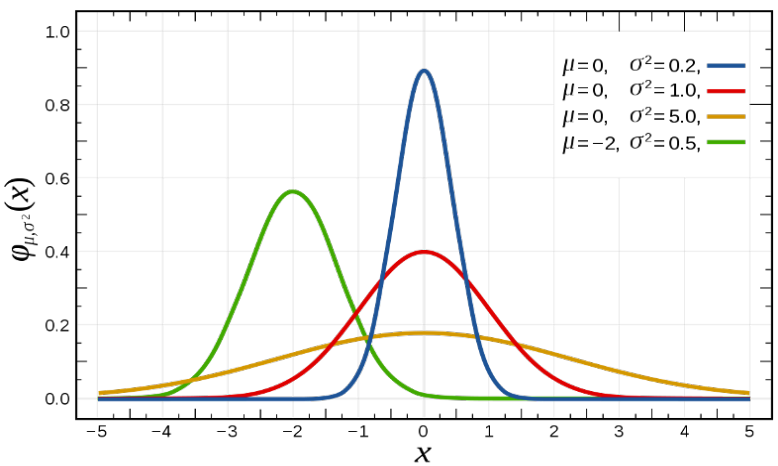
- construct confidence sets for unknown expected rewards
- select arms according to their highest upper confidence bounds
- regret upper bound of order $O\left(\frac{\log T}{\Delta}\right)$ [1]



[1] $\Delta = \min_{i \in [m]} \{\mu^* - \mu_i\}$

Thompson sampling (TS)

- maintain an iteratively updated posterior distribution for unknown expected rewards
- select arms according to probabilities of being the best one
- was introduced in 1933[2], but has not been theoretically proven until recent years[3]
- regret upper bound of order $O\left(\frac{\log T}{\Delta}\right)$



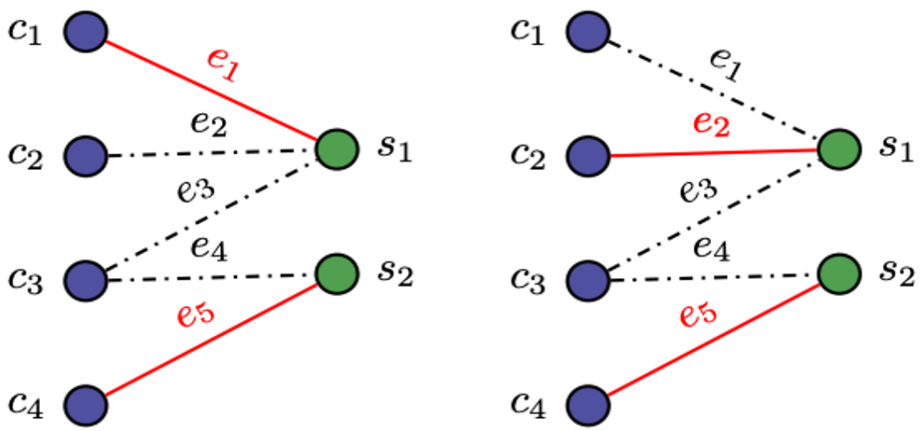
[2] William R Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3/4 (1933), pp. 285–294.

[3] Shipra Agrawal and Navin Goyal. "Analysis of thompson sampling for the multi-armed bandit problem". In: *Conference on Learning Theory*. 2012, pp. 39–1.

Combinatorial multi-armed bandit (CMAB)

Advertisement placement

- The agent needs to select several webpages (left) to place the advertisement, thus the click-through rate is maximized when faced with several users (right).
- can be formulated by probabilistic maximum coverage (PMC)



Combinatorial multi-armed bandit (CMAB)

- for round $t = 1, 2, \ldots$:
- the player selects a combination of arms: S_t satisfying $S_t \in \mathcal{C}$
- the environment samples the outcome $X_{t,i}$ for each arm i
- the player observes the feedback $Q_t = \{(i, X_{t,i}) : i \in S_t\}$ and receives the reward $R(S_t, X_t)$
- The goal of the player is to maximize the cumulative expected reward

$$\mathbb{E}_{S_t, X_t}\left[\sum_{t=1}^{T} R(S_t, X_t)\right]$$

Assumption

The expected reward of an action S only depends on the S and the mean vector μ. That is to say, there exists a function r such that $\mathbb{E}[R_t] = \mathbb{E}_{X_t \sim D}[R(S_t, X_t)] = r(S_t, \mu)$.

Assumption

(Lipschitz continuity) There exists a constant B such that for any action S and mean vectors μ, μ', the reward of S under μ and μ' satisfies

$$\left|r(S,\mu) - r(S,\mu')\right| \leq B \sum_{i \in S} \left|\mu_i - \mu_i'\right| .$$

Combinatorial multi-armed bandit (CMAB)

CMAB algorithms usually rely on an oracle to solve the corresponding offline problem.

- when the expected rewards are known, the problem of finding the optimal solution $\text{argmax}_{S \in \mathcal{C}} r(S, \mu)$ is called *offline* problem
- many offline problems are NP-hard (such as maximum coverage, influence maximization) and only approximate algorithms (oracles) are available
- an Oracle is (α, β)-approximate if $r(\mathit{Oracle}(\mu'), \mu') \geq \alpha \max_{S \in \mathcal{C}} r(S, \mu')$ with probability larger than β for any given μ'

Combinatorial multi-armed bandit (CMAB)

With an (α, β)-approximate oracle, the goal is equivalent to minimize the (α, β)-approximate regret

$$\mathbb{E}\left[T \cdot \alpha\beta \cdot r(S^*, \mu) - \sum_{t=1}^{T} r(S_t, \mu)\right]$$

Related work: two popular algorithms

upper confidence bound - type (UCB)

- construct confidence sets for unknown expected reward of each arm
- an oracle helps to choose the action according to arms' highest upper confidence bounds
- require the reward function to satisfy the monotonicity on μ

Algorithm type	allow approximate oracle	regret guarantee[4]
UCB-type	yes, $\forall \alpha, \beta \in (0, 1]$	$O\left(\frac{\log T}{\Delta}\right)^5$

[4] $\Delta = \min_{S \in \mathcal{C}: S \notin \alpha OPT} \{\alpha \cdot r(S^*, \mu) - r(S, \mu)\}$

[5] Qinshi Wang and Wei Chen. "Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1161–1171.

Related work: two popular algorithms

Thompson sampling - type (TS)

- maintain an iteratively updated posterior distribution for unknown expected reward of each arm
- an oracle helps to choose the action according to sampled parameters $(\theta_i)_{i\in[m]}$ from posterior distributions
- relax the monotonicity requirment of UCB-type algorithms
- easier implementation and better empirical performance

Algorithm type	allow approximate oracle	regret guarantee[6]
TS-type	no, $\alpha = \beta = 1$	$O\left(\frac{\log T}{\Delta}\right)$[7]

[6] $\Delta = \min_{S\in\mathcal{C}:S\notin OPT}\{r(S^*,\mu) - r(S,\mu)\}$

[7] Pierre Perrault et al. "Statistical Efficiency of Thompson Sampling for Combinatorial Semi-Bandits". In: *Advances in Neural Information Processing Systems*. 2020, Siwei Wang and Wei Chen. "Thompson sampling for combinatorial semi-bandits". In: *International Conference on Machine Learning*. PMLR, 2018, pp. 5114–5122.

Motivation

An example [Theorem 2, Work[8]] shows the failure of TS to learn with an approximation oracle.

But this oracle is uncommon and is designed only for a specific problem instance.

Can the convergence analysis of TS extend beyond the exact oracle in CMAB ?

[8] Siwei Wang and Wei Chen. "Thompson sampling for combinatorial semi-bandits". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5114–5122.

Setting

We consider TS with common Greedy oracle, which can provide approximate solutions for many offline problems

- m base arms, denoted by $[m] = \{1, 2, \ldots, m\}$
- due to the problem structure, arms are further divided into n units, a unit of arms must be selected together. Denote the unit set as $\mathcal{U}$
- for round $t = 1, 2, \ldots$
- the player selects action $S_t \in \mathcal{S} = \{S \subseteq \mathcal{U} : |S| = K\}$
- the environment samples the outcome $X_{t,i}, \forall i \in [m]$
- the player observes the feedback $Q_t = \{(i, X_{t,i}) : i \in s \text{ for some } s \in S_t\}$
- the player receives the reward $R(S_t, X_t)$

Greedy oracle

Algorithm 1 Greedy algorithm

1: Input: base arm set $[m]$ and mean vector $\mu = (\mu_i)_{i\in[m]}$, unit set $\mathcal{U}$, action size K
2: Initialize: $S_{g,0} = \emptyset$
3: **for** $k = 1, 2, \cdots, K$ **do**
4: $\quad s_k = \text{argmax}_{s\in\mathcal{U}\setminus S_{g,k-1}} r(S_{g,k-1} \cup \{s\}, \mu)$
5: $\quad S_{g,k} = S_{g,k-1} \cup \{s_k\}$
6: **end for**
7: Output: $S_g = S_{g,K}$

The above framework with Greedy oracle can cover many CMAB problems including probabilistic maximum coverage (PMC).

For simplicity, we assume S_g is unique under μ.

Combinatorial Thompson sampling (CTS)

Algorithm 2 CTS algorithm with Greedy oracle

Input: base arm set $[m]$, unit set $\mathcal{U}$, action size K
Initialize: $\forall i \in [m]$, G_i is the prior distribution
for $t = 1, 2, \cdots$ **do**
    $\forall i \in [m]$: Sample $\theta_{t,i} \sim G_i$.
    Select action $S_t = Greedy([m], \theta_t, \mathcal{U}, K)$ and receive the observation Q_t
    //update
    for $(i, X_{t,i}) \in Q_t$ **do**
        Update the posterior distribution G_i
    end for
end for

Objective: α-approximate regret?

under UCB:

$$\begin{aligned}\alpha \cdot r(S^*, \mu) - r(S_t, \mu) &\leq \alpha \cdot r(S^*, U) - r(S_t, \mu) \\ &\leq r(S_t, U) - r(S_t, \mu) \\ &\leq \sum_{i \in S_t} |U_i - \mu_i| .\end{aligned}$$

under TS:

$$\begin{aligned}&\alpha \cdot r(S^*, \mu) - r(S_t, \mu) \\ =\ &\alpha \cdot r(S^*, \mu) - \alpha \cdot r(S^*, \theta) + \alpha \cdot r(S^*, \theta) - r(S_t, \mu) \\ \leq\ &\alpha \cdot (r(S^*, \mu) - r(S^*, \theta)) + r(S_t, \theta) - r(S_t, \mu) \\ \leq\ &\alpha \sum_{i \in S^*} |\theta_i - \mu_i| + \sum_{i \in S_t} |\theta_i - \mu_i| .\end{aligned}$$

It is first brought up in analyzing UCB-based algorithms and may not well fit TS-based algorithms.

Objective

A different objective to minimize the greedy regret

$$R_g(T) = \mathbb{E}\left[\sum_{t=1}^{T} \max\{r(S_g, \mu) - r(S_t, \mu), 0\}\right]. \tag{2}$$

Remark

When Greedy can provide α-approximate solutions, the upper bound for the greedy regret also implies the upper bound for the α-approximate regret since $r(S_g, \mu) \geq \alpha \cdot r(S^*, \mu)$.

Lower bound: define gaps to quantify the hardness

For any unit $s \in \mathcal{U}$ and index $k \in [K]$ such that $s \notin S_{g,k}$, define the marginal reward gap

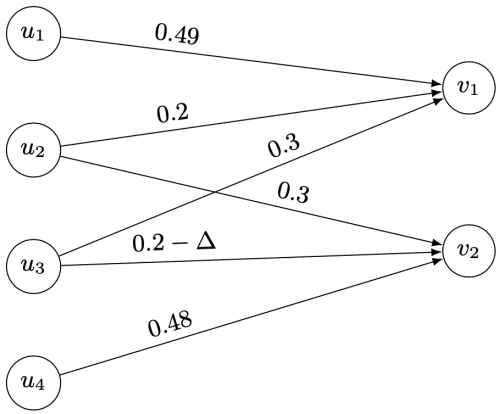
$$\Delta_{s,k} = r(S_{g,k}, \mu) - r(S_{g,k-1} \cup \{s\}, \mu)$$

as the reward difference between $S_{g,k}$ and $S_{g,k-1} \cup \{s\}$. According to the Greedy algorithm, we have $\Delta_{s,k} > 0$ for any k such that $s \notin S_{g,k}$. And for any action $S \in \mathcal{S}$, define $\Delta_S = \max\{r(S_g, \mu) - r(S, \mu), 0\}$ as the reward difference from the Greedy's solution S_g. Let

$$\Delta_s^{\min} = \min_{S \in \mathcal{S}: s \in S} \Delta_S, \quad \Delta_s^{\max} = \max_{S \in \mathcal{S}: s \in S} \Delta_S$$

be the minimum and maximum reward gap of actions containing unit s, respectively.

Lower Bound: problem instance



Action	Expected Reward	Action	Expected Reward
$\{u_1\}$	0.49	$\{u_1, u_2\}$	0.892
$\{u_2\}$	0.5	$\{u_1, u_3\}$	$0.843 - \Delta$
$\{u_3\}$	$0.5 - \Delta$	$\{u_1, u_4\}$	0.97
$\{u_4\}$	0.48	$\{u_2, u_3\}$	$0.88 - 0.7\Delta$
$\{u_3, u_4\}$	$0.884 - 0.52\Delta$	$\{u_2, u_4\}$	0.836

Figure: The underlying graph of the PMC instance used to derive the hardness analysis and the corresponding rewards of actions.

$S_g = \{u_2, u_1\}$ with $s_{g,1} = u_2$, $s_{g,2} = u_1$, and $r(S_g, \mu) = 0.892$, while the optimal action is $\{u_1, u_4\}$

Lower bound: intuition and result

Using the CTS algorithm with Gaussian priors and Greedy oracle to solve the CMAB problem shown in Figure 1, when T is sufficiently large, we have

$$\mathbb{E}\left[N_{T+1,s}\right] = \Omega\left(\frac{\log T}{\Delta_{s,1}^2}\right), \tag{3}$$

for any $s \neq s_{g,1} = u_2$, where $N_{T+1,s} = \sum_{t=1}^{T} \mathbb{1}\{s \in S_t\}$ is the number of rounds when s is contained in the selected action set S_t.

Lower bound: intuition and result

$$\Delta_{u_3,1} = \Delta, \quad \Delta_{u_3,2} = 0.012 + 0.7\Delta\,, \quad \Delta_{u_3}^{\min} = 0.52\Delta + 0.008\,.$$

Theorem

(Lower bound) When T is sufficiently large, the cumulative greedy regret satisfies

$$R_g(T) = \Omega\left(\frac{\log T}{\Delta_{u_3,1}^2} \cdot \Delta_{u_3}^{\min}\right) = \Omega\left(\frac{\log T}{\Delta^2}\right)\,. \tag{4}$$

Upper bound

Theorem

(Upper bound) The cumulative greedy regret of CTS with Gaussian and Beta priors can be upper bounded by

$$R_g(T) = O\left(\sum_{s \neq s_{g,1}} \max_{k: s \notin S_{g,k}} \frac{B^2 |s|^2 \Delta_s^{\max} \log T}{\Delta_{s,k}^2}\right), \tag{5}$$

where B is the coefficient of the Lipschitz continuity condition, $|\cup S_g|$ is the number of base arms that belong to the units contained in S_g.

Upper bound: Comparisons with TS for MAB

- $K = 1$, $B = 1$ and $|s| = 1$ for any s
- Greedy can provide exact optimal solutions thus $R_g(T) = R(T) = \mathbb{E}\left[\sum_{t=1}^{T}(\mu^* - \mu_{A_t})\right]$
- $\Delta_s^{\max} = \Delta_{s,1}$ for any s
- Our greedy regret upper bound is

$$R_g(T) = O\left(\sum_{s \neq s_{g,1}} \frac{\log T}{\Delta_{s,1}}\right),$$

which recovers the main order of the regret of TS for MAB[9].

[9] Shipra Agrawal and Navin Goyal. "Further optimal regret bounds for thompson sampling". In: *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*. 2013, pp. 99–107.

Upper bound: proof sketch

$$
\begin{array}{lccccc}
S_g = & \{s_{g,1}, & & s_{g,2}, & \dots & s_{g,K}\} \\
S_t = & \{s_{t,1}, & \rightarrow & s_{t,2}, & \dots & s_{t,K}\} \\
 & s_{t,1} = s_{g,1}? & \rightarrow & s_{t,2} = s_{g,2}? \mid s_{t,1} = s_{g,1} & & s_{t,K} = s_{g,K}? \mid S_{t,K-1} = S_{g,K-1} \\
 & \forall s \neq s_{g,1} : \frac{\log T}{\Delta^2_{s,1}} & & \forall s \notin S_{g,2} : \frac{\log T}{\Delta^2_{s,2}} & & \forall s \notin S_{g,K} : \frac{\log T}{\Delta^2_{s,K}}
\end{array}
$$

Above all, each unit s needs to be explored

$$
O\left(\max_{k:s\notin S_{g,k}} \frac{\log T}{\Delta^2_{s,k}}\right)
$$

Conclusion

- The first theoretical results for TS-type algorithm to solve CMAB problems with (approximate) greedy oracle.
- Our results break the misconception that CTS cannot be used with approximation oracles with the failure example in work[10].

[10] Siwei Wang and Wei Chen. "Thompson sampling for combinatorial semi-bandits". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5114–5122.

Future work

An interesting future direction is to extend the current CMAB framework to the case with probabilistically triggered arms (CMAB-T).

- the CMAB-T framework models more applications
- TS samples candidate parameters to escape the computation of complicated optimization problems which may be faced in UCB[11]
- new proof techniques are required

[11] Shuai Li et al. "Online Influence Maximization under Linear Threshold Model". In: *Advances in Neural Information Processing Systems*. 2020.

Thanks!
Questions?