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

Fang Kong, Yueran Yang, Wei Chen, Shuai Li

上海交通大学 约翰·霍普克罗夫特计算机科学中心 John Hopcroft Center for Computer Science

Background and Motivation

- Combinatorial multi-armed bandits (CMAB)
  - A sequential decision-making problem
  - The agent selects a combination of base arms as an action to play in each round, and all outcomes of these selected arms are then revealed (semi-bandit feedback).
  - An offline oracle helps to find the solution in each round with estimated parameters as input.
  - Many applications: probabilistic maximum coverage (PMC), online influence maximization (OIM), multiple-play MAB (MP-MAB), minimum spanning tree (MST)
- Thompson Sampling
  - It was introduced very earlier in the 1930s.
  - Compared with UCB-type algorithms, TS-type algorithms do not require the reward function to satisfy the monotonicity on the mean vector of base arms and also benefit from other advantages of easier implementation and better practical performances.
  - Previous TS-based analyses all require an exact oracle to solve CMAB problems.
- However, exact oracles are usually not feasible in CMAB problems since many offline combinatorial optimization problems are NP-hard and only approximation oracles are available. With an example [WC18] illustrating the non-convergent regret of TS with an artificial approximation oracle designed for a specific problem instance, whether TS can work well in CMAB problems with common approximation oracles is still an open problem.
- Greedy oracle is common for offline combinatorial optimization problems.
  - It can provide approximate solutions for offline problems of PMC and OIM, and exact optimal solutions for offline problems of MP-MAB and MST.
  - In general, as long as the expected reward in a problem satisfies the monotonicity and submodularity on the action set, the greedy algorithm serves as an offline oracle to provide an approximate solution [NWF78].

Setting

- Problem Formulation
  - m base arms and the arm set is denoted by $[m] = \{1, 2, \ldots, m\}$. Each arm $i \in [m]$ is associated with a distribution D_i on $[0, 1]$
  - The base arm set $[m]$ is further divided into n units, with each unit containing several base arms and a unit of arms will be selected together.
  - Let $\mathcal{U}$ be the collection of all units and $|s|$ be the number of base arms contained in unit s for any $s \in \mathcal{U}$.
- The online problem, in each round t:
  - The learning agent selects an action $S_t \in \mathcal{S} = \{S \subseteq \mathcal{U} : |S| = K\}$ to play. $\mathcal{S}$ is the set of all candidate actions containing K units and $\cup S = \{i \in s \text{ for some } s \in S\}$ is the set of base arms that belong to units contained in S.
  - The environment then draws a random output of all base arms $X_t = (X_{t,1}, X_{t,2}, \ldots, X_{t,m})$ from the distribution $D = D_1 \times D_2 \times \ldots \times D_m$. For any t, $X_{t,i}$ is independent and identically distributed on D_i with expectation μ_i.
  - The agent can observe feedback $Q_t = \{(i, X_{t,i}) \mid i \in \cup S_t\}$, namely the output of all base arms in units contained in S_t.
  - The agent finally obtains a corresponding reward $R_t = R(S_t, X_t)$ in this round, which is a function of action S_t and output X_t and satisfies the following widely-studied assumptions.

    Assumption 1. *The expected reward of an action S only depends on 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 2. *(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*

$$\left|r(S, \mu) - r(S, \mu')\right| \leq B \sum_{i \in \cup S} \left|\mu_i - \mu_i'\right| . \tag{1}$$

- The offline problem, `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**: $S_g = \emptyset$
  3. **for** $k = 1, 2, \cdots, K$
  4. $\quad s_k = \operatorname{argmax}_{s \in \mathcal{U} \setminus S_g} r(S_g \cup \{s\}, \mu)$
  5. $\quad S_g = S_g \cup \{s_k\}$
  6. **Output:** Output: S_g
- Objective
  - To simplify, we first assume the `Greedy`'s solution $S_g(\mu)$, abbreviated as S_g, is unique, or equivalently the optimal unit in each step k is unique.
  - The objective of the learning agent is to minimize the cumulative expected regret with respect to the `Greedy`'s solution S_g, which we call cumulative *greedy regret* defined by

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

  where the expectation is taken from the randomness in observations and the online algorithm.

Remark. *When `Greedy` is α-approximate, the upper bound for greedy regret also implies the upper bound for the α-approximate regret which is widely adopted in previous CMAB works based on UCB-type algorithms [WC17]. The α-approximate regret is weaker than greedy regret as it relaxes the requirements for online algorithms and only needs them to return solutions satisfying the relaxed approximation ratio.*

Combinatorial Thompson sampling (CTS) algorithm

CTS algorithm with Beta priors and `Greedy` oracle

1. **Input:** base arm set $[m]$, unit set $\mathcal{U}$, action size K
2. **Initialize**: $\forall i \in [m], a_i = b_i = 1$
3. **for** $t = 1, 2, 3, \ldots$
4. $\quad \forall i \in [m]$: Sample $\theta_{t,i} \sim$Beta(a_i, b_i). Denote $\theta_t = (\theta_{t,1}, \theta_{t,2}, \cdots, \theta_{t,m})$
5. $\quad$ Select action $S_t =$ `Greedy`$([m], \theta_t, \mathcal{U}, K)$ and receive the observation Q_t
6. $\quad$ //Update
7. $\quad$ **for** $(i, X_{t,i}) \in Q_t$
8. $\quad\quad$ With probability $X_{t,i}$, $Y_{t,i} = 1$; with probability $1 - X_{t,i}$, $Y_{t,i} = 0$
9. $\quad\quad$ Update $a_i = a_i + Y_{t,i}, b_i = b_i + (1 - Y_{t,i})$

Main Results

- Definitions
  - $S_g = \left\{s_{g,1}, s_{g,2}, \ldots, s_{g,K}\right\}$, where $s_{g,k}$ is the k-th selected unit by `Greedy`.
  - $S_{g,k} = \left\{s_{g,1}, s_{g,2}, \ldots, s_{g,k}\right\}$ is the sequence containing the first k units for any $k \in [K]$.
  - Similarly, let $S_t = \left\{s_{t,1}, s_{t,2}, \ldots, s_{t,K}\right\}$ and $S_{t,k} = \left\{s_{t,1}, s_{t,2}, \ldots, s_{t,k}\right\}$.
  - (Gaps) For any unit $s \in \mathcal{U}$ and index $k \in [K]$ such that $s \notin S_{g,k-1}$, 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 $\Delta_S = \max\left\{r(S_g, \mu) - r(S, \mu), 0\right\}$ 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. Denote $\Delta_{\max} = \max_{S \in \mathcal{S}} \Delta_S$ as the maximum reward gap over all suboptimal actions.

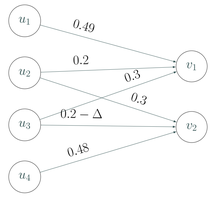

Figure 1: The underlying graph of the PMC instance used to derive the lower bound.

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

Table 1: The expected rewards of actions in the problem instance shown in Figure 1.

Theorem 1. *(Lower bound) 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.

Further, the cumulative greedy regret satisfies

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

Theorem 2. *(Upper bound) The cumulative greedy regret of CTS Algorithm with `Greedy` oracle and Beta (or Gaussian) priors can be upper bounded by*

$$R_g(T) \leq O\left(\sum_{s \neq s_{g,1}} \max_{k: s \notin S_{g,k}} \frac{B^2 |s|^2 \Delta_s^{\max} \log T}{\left(\Delta_{s,k} - 2B\left|\cup S_g\right| \varepsilon\right)^2} + \sum_{k \in [K]} \frac{C}{\varepsilon^2} \left(\frac{C'}{\varepsilon^4}\right)^{|s_{g,k}|} \Delta_{\max}\right) \tag{5}$$

for any ε such that $\forall s \neq s_{g,1}$ and k satisfying $s \notin S_{g,k}$, $\Delta_{s,k} > 2B\left|\cup S_g\right| \varepsilon$, where B is the coefficient of the Lipschitz continuity condition, $\left|\cup S_g\right|$ is the number of base arms that belong to the units contained in S_g, C and C' are two universal constants.

Conclusion and Future Work

- We give the first theoretical result for TS-type algorithm to solve CMAB problems with (approximate) greedy oracle.
- Our result breaks the misconception that CTS cannot be used with approximation oracles.
- An interesting future direction is to extend the current CMAB framework to the case with probabilistically triggered arms (CMAB-T).

References

[NWF78] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[WC17] 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*, pages 1161–1171, 2017.

[WC18] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *Proceedings of the 35th International Conference on International Conference on Machine Learning*, pages 5114–5122, 2018. https://arxiv.org/abs/1803.04623.