Combinatorial Multi-Armed Bandit: General Framework, Results and Applications

Wei Chen WEIC@MICROSOFT.COM
Microsoft Research Asia, Beijing, China

Yajun Wang YAJUNW@MICROSOFT.COM
Microsoft Research Asia, Beijing, China

Yang Yuan YANGYUAN@CS.CORNELL.EDU
Computer Science Department, Cornell University, Ithaca, NY, USA

Abstract

We define a general framework for a large class of combinatorial multi-armed bandit (CMAB) problems, where simple arms with unknown distributions form *super arms*. In each round, a super arm is played and the outcomes of its related simple arms are observed, which helps the selection of super arms in future rounds. The reward of the super arm depends on the outcomes of played arms, and it only needs to satisfy two mild assumptions, which allow a large class of nonlinear reward instances. We assume the availability of an (α, β)-approximation oracle that takes the means of the distributions of arms and outputs a super arm that with probability β generates an α fraction of the optimal expected reward. The objective of a CMAB algorithm is to minimize (α, β)-*approximation regret*, which is the difference in total expected reward between the $\alpha\beta$ fraction of expected reward when always playing the optimal super arm, and the expected reward of playing super arms according to the algorithm. We provide CUCB algorithm that achieves $O(\log n)$ regret, where n is the number of rounds played, and we further provide distribution-independent bounds for a large class of reward functions. Our regret analysis is tight in that it matches the bound for classical MAB problem up to a constant factor, and it significantly improves the regret bound in a recent paper on combinatorial bandits with linear rewards. We apply our CMAB framework to two new applications, probabilistic maximum coverage (PMC) for online advertising and social influence maximization for viral marketing, both having nonlinear reward structures.

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

1. Introduction

Multi-armed bandit (MAB) is a problem extensively studied in statistics and machine learning. The classical version of the problem is formulated as a system of m arms (or machines), each having an unknown distribution of the reward with an unknown mean. The task is to repeatedly play these arms in multiple rounds so that the total expected reward is as close to the reward of the optimal arm as possible. An MAB algorithm needs to decide which arm to play in the next round given the outcomes of the arms played in the previous rounds. The metric for measuring the effectiveness of an MAB algorithm is its *regret*, which is the difference in the total expected reward between always playing the optimal arm and playing arms according to the algorithm. The MAB problem and its solutions reflect the fundamental tradeoff between exploration and exploitation: whether one should try some arms that have not been played much (exploration) or one should stick to the arms that provide good reward so far (exploitation). Existing results show that one can achieve a regret of $O(\log n)$ when playing arms in n rounds, and this is asymptotically the best.

In many real-world applications, the setting is not the simple MAB one, but has a combinatorial nature among multiple arms and possibly non-linear reward functions. For example, consider the following online

advertising scenario. A web site contains a set of web pages and has a set of users visiting the web site. An advertiser wants to place an advertisement on a set of selected web pages on the site, and due to his budget constraint, he can select at most k web pages. Each user visits a certain set of pages, and on each visited page has a certain click-through probability of clicking the advertisement on the page, but the advertiser does not know these probabilities. The advertiser wants to repeatedly select sets of k web pages, observe the click-through data collected to learn the click-through probabilities, and maximize the number of users clicking his advertisement. Another example is viral marketing in online social networks, where a marketer repeatedly selects seed nodes in a social network, observes the cascading behavior of the viral information to learn influence probabilities between individuals in the social network, with the goal of maximizing the overall effectiveness of all viral cascades.

In the above examples, page-user pairs or pairs of nodes in the social network can be viewed as arms, but they are not played one by one. Instead, these arms form certain combinatorial structures (e.g. bipartite graphs in the online advertising scenario and directed graphs in the viral marketing scenario), and in each round, a set of arms (called a *super arm*) are played together. Moreover, the reward structure is not a simple linear function of the outcomes of all played arms but takes a more complicated form. For example, in the online advertising scenario, for all page-user pairs with the same user, the collective reward of these arms is either 1 if the user clicks the advertisement on at least one of the pages and 0 if the user does not click the advertisement on any page.

It is possible to treat every super arm as an arm and simply apply the classical MAB framework to solve the above combinatorial problems. However, such naive treatment has two issues. First, the number of super arms may be exponential to the problem instance size due to combinatorial explosion, and thus classical MAB algorithms may need exponential number of steps just to go through all the super arms. Second, after one super arm is played, in many cases, we can observe some information regarding the outcomes of the underlying arms, which may be shared by other super arms. However, this information is discarded in the classical MAB framework, making it less effective.

In this paper, we define a general framework for the *combinatorial multi-armed bandit (CMAB)* problem to address the above issues and cover a large class of combinatorial online learning problems (Section 2). In the CMAB framework, a super arm is a set of underlying arms, whose outcomes follow unknown distributions. In each round one super arm is played and the outcomes of all arms in the super arm (and possibly some other triggered arms) are revealed. A CMAB algorithm needs these information from the past rounds to decide the super arm to play in the next round.

The framework allows an arbitrary combination of arms into super arms. The reward function only needs to satisfy two mild assumptions, and thus covering a large class of nonlinear reward functions. We do not assume the direct knowledge on how super arms are formed from underlying arms or how the reward is computed. Instead, we assume the availability of an offline computation oracle that takes such knowledge as well as the expectations of outcomes of all arms as input and computes the optimal super arm with respect to the input. Since many combinatorial problems are computationally hard, we further allow approximation oracles with failure probabilities. In particular, we relax the oracle to be an (α, β)-approximation oracle for some $\alpha, \beta \leq 1$, that is, with success probability β, the oracle could output a super arm whose expected reward is at least α fraction of the optimal expected reward. As a result, our regret metric is not comparing against the expected reward of playing the optimal super arm each time, but against the $\alpha\beta$ fraction of the optimal expected reward, since the offline oracle can only guarantee this fraction in expectation. We refer to this as the (α, β)-*approximation regret*.

For the general framework, we provide the CUCB (combinatorial upper confidence bound) algorithm (Section 3), an extension to the UCB1 algorithm for the classical MAB problem (Auer et al., 2002a). We prove that the regret of CUCB is bounded by $O(\log n)$. Our regret analysis is tight in that when applying it to the classical MAB problem we obtain a regret bound that matches the bound of the classical MAB up to a constant factor. Our tight analysis further allows us to provide a distribution-independent regret bound that works for arbitrary distributions of underlying arms, for a large class of CMAB instances.

We then apply our general framework and provide solutions to two new bandit applications, the probabilistic maximum coverage problem for advertisement placement and social influence maximization for viral marketing (Section 4). The offline version of both problems are NP-hard, with constant approximation algorithms available. Both problems have nonlinear reward structures that cannot be handled by any existing work. The social influence maximization problem provides an interesting instance in which playing one super arm not only reveals the outcomes of the under-

lying arms it contains, but may stochastically trigger more arms to reveal their outcomes, and the reward depends on the outcomes of all revealed arms.

We also apply our result to combinatorial bandits with linear rewards, recently studied in (Gai et al., 2012) (Section 4.3). We show that we significantly improve their regret bound, even though we are covering a much larger class of combinatorial bandit instances.

In the supplementary material, besides providing the full proof of our main theorems, we further provide (a) an ε_t-greedy algorithm for CMAB, and (b) an improved regret analysis for CMAB where arms are clustered and played together, which can be applied to the two new applications studied in this paper.

In summary, our contributions include: (a) defining a general CMAB framework that encompasses a large class of nonlinear reward functions, (b) providing CUCB algorithm with a tight regret analysis as a general solution to this CMAB framework, and (c) demonstrating that our general framework can be effectively applied to a number of practical combinatorial bandit problems, including ones with nonlinear rewards. Moreover, our framework provides a clean separation of the online learning task and the offline computation task: the oracle takes care of the offline computation task, which uses the domain knowledge of the problem instance, while our CMAB algorithm takes care of the online learning task, and is oblivious to the domain knowledge of the problem instance.

Related work. Multi-armed bandit problem has been well studied in the literature, in particular in statistics and reinforcement learning (cf. (Berry & Fristedt, 1985; Sutton & Barto, 1998)). Our work follows the line of research on stochastic MAB problems, which is initiated by Lai and Robbins (Lai & Robbins, 1985), who show that under certain conditions on reward distributions, one can achieve a tight asymptotic regret of $\Theta(\log n)$, where n is the number of rounds played. Later, Auer et al. demonstrate that $O(\log n)$ regret can be achieved uniformly over time rather than only asymptotically (Auer et al., 2002a). They propose several MAB algorithms, including the UCB1 algorithm, which has been widely followed and adapted in MAB research.

For combinatorial multi-armed bandits, a few specific instances of the problem has been studied in the literature. A number of studies consider simultaneous plays of k arms among m arms (e.g. (Anantharam et al., 1987a; Caro & Gallien, 2007; Liu et al., 2011)). Other instances include the matching bandit (Gai et al., 2010) and the online shortest path problem (Liu & Zhao, 2012).

The work closest to ours is a recent work by Gai et al. (Gai et al., 2012), which also considers a combinatorial bandit framework with an approximation oracle. However, our work differs from theirs in several important aspects. Most importantly, their work only considers linear rewards while our CMAB framework includes a much larger class of linear and nonlinear rewards. Secondly, our regret analysis is much tighter, and as the result we significantly improve their regret bound when applying our result to the linear reward case, and we are able to derive a distribution-independent regret bound while their results cannot lead to distribution-independent bounds. Moreover, we allow the approximation oracle to have a failure probability (i.e., $\beta < 1$), which they do not consider.

In terms of types of feedbacks in combinatorial bandits (Audibert et al., 2011), our work belongs to the *semi-bandit* type, in which the player observes only the outcomes of played arms in one round of play. Other types include (a) *full information*, in which the player observes the outcomes of all arms, and (b) *bandit*, in which the player only observes the final reward but no outcome of any individual arm. More complicated feedback dependences are also considered in (Mannor & Shamir, 2011).

A different line of research considers *adversarial multi-armed bandit*, initiated by the work in (Auer et al., 2002b), in which an adversary controls the arms and tries to defeat the learning process. In the context of adversarial bandits, several studies also consider combinatorial bandits (Cesa-Bianchi & Lugosi, 2009; Audibert et al., 2011; Bubeck et al., 2012). For linear rewards, Kakade et al. (Kakade et al., 2009) have shown how to convert an approximation oracle into an online algorithm with sublinear regret both in the full information setting and the bandit setting. For nonlinear rewards, various online submodular optimization problems with bandit feedback are studied in the adversarial setting (Streeter & Golovin, 2008; Radlinski et al., 2008; Streeter et al., 2009; Hazan & Kale, 2009). Notice that our framework deals with stochastic instances and we can handle reward functions more general than the submodular ones.

2. General CMAB Framework

A CMAB problem consists of m arms associated with a set of random variables $X_{i,t}$ for $1 \leq i \leq m$ and $t \geq 1$, with bounded support on $[0, 1]$. Variable $X_{i,t}$ indicates the random outcome of the i-th arm in its t-th trial. The set of random variables $\{X_{i,t} \mid t \geq 1\}$ associated with arm i are independent and identically dis-

tributed according to some unknown distribution with unknown expectation μ_i. Let $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_m)$ be the vector of expectations of all arms. Random variables of different arms may be dependent.

The CMAB problem contains a constraint $\mathcal{S} \subseteq 2^{[m]}$, where $2^{[m]}$ is the set of all possible subsets of arms. We refer to every set of arms $S \in \mathcal{S}$ as a *super arm* of the CMAB problem. In each round, one super arm $S \in \mathcal{S}$ is played and the outcomes of arms in S are revealed. More precisely, for each arm $i \in [m]$, let $T_{i,t}$ denote the number of times the outcome of arm i is revealed after the first t rounds in which t super arms are played. If $S \in \mathcal{S}$ is the super arm played in round t, the outcomes of random variables $X_{i,T_{i,t}}$ for all $i \in S$ are revealed. For some problem instances (e.g. social influence maximization in Section 4.2), the outcomes of other arms may also be revealed depending on the outcomes of arms in S.

Let $R_t(S)$ be a non-negative random variable denoting the reward of round t when super arm S is played. The reward depends on the actual problem instance definition, the super arm S played, and the outcomes of the revealed arms in round t. The reward $R_t(S)$ might be as simple as a summation of the outcomes of the arms in S: $R_t(S) = \sum_{i \in S} X_{i,T_i(t)}$, but our framework allows more sophisticated nonlinear rewards, as explained below.

In this paper, we consider CMAB problems in which the expected reward of playing any super arm S in any round t, $\mathbb{E}[R_t(S)]$, is a function of only the set of arms S and the expectation vector $\boldsymbol{\mu}$ of all arms. For the linear reward case as given above, this is true because linear addition is commutative with the expectation operator. For non-linear reward functions not commutative with the expectation operator, it is still true if we know the type of distributions and only the expectations of arm outcomes are unknown, and outcomes of different arms are independent. For example, the distribution of $X_{i,t}$'s are known to be 0-1 Bernoulli random variables with unknown mean μ_i. Henceforth, we denote the expected reward of playing S as $r_{\boldsymbol{\mu}}(S) = \mathbb{E}[R_t(S)]$. To carry out our analysis, we make the following two mild assumptions on the expected reward $r_{\boldsymbol{\mu}}(S)$:

- **Monotonicity**. The expected reward of playing any super arm $S \in \mathcal{S}$ is monotonically nondecreasing with respect to the expectation vector, i.e., if for all $i \in [m]$, $\mu_i \le \mu'_i$, we have $r_{\boldsymbol{\mu}}(S) \le r_{\boldsymbol{\mu}'}(S)$ for all $S \in \mathcal{S}$.
- **Bounded smoothness.** There exists a strictly increasing (and thus invertible) function $f(\cdot)$, called *bounded smoothness function*, such that for any two expectation vectors $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$, we have $|r_{\boldsymbol{\mu}}(S) - r_{\boldsymbol{\mu}'}(S)| \le f(\Lambda)$ if $\max_{i \in S} |\mu_i - \mu'_i| \le \Lambda$.

Both assumptions are natural. In particular, they hold true for all the applications we considered.

A CMAB algorithm A is one that selects the super arm of round t to play based on the outcomes of revealed arms of previous rounds, without knowing the expectation vector $\boldsymbol{\mu}$. Let S_t^A be the super arm selected by A in round t. Note that S_t^A is a random super arm that depends on the outcomes of arms in previous rounds and potential randomness in the algorithm A itself. The objective of algorithm A is to maximize the expected reward of all rounds up to a round n, that is, $\mathbb{E}_{S,R}[\sum_{t=1}^{n} R_t(S_t^A)] = \mathbb{E}_S[\sum_{t=1}^{n} r_{\boldsymbol{\mu}}(S_t^A)]$, where $\mathbb{E}_{S,R}$ denotes taking expectation among all random events generating the super arms S_t^A's and generating rewards $R_t(S_t^A)$'s, and $\mathbb{E}_S$ denotes taking expectation only among all random events generating the super arms S_t^A's.

We do not assume that the learning algorithm has the direct knowledge about the problem instance, e.g. how super arms are formed from the underlying arms and how reward is defined. Instead, the algorithm has access to a computation oracle that takes the expectation vector $\boldsymbol{\mu}$ as the input, and together with the knowledge of the problem instance, computes the optimal or near-optimal super arm S. Let $\text{opt}_{\boldsymbol{\mu}} = \max_{S \in \mathcal{S}} r_{\boldsymbol{\mu}}(S)$ and $S^*_{\boldsymbol{\mu}} = \text{argmax}_{S \in \mathcal{S}} r_{\boldsymbol{\mu}}(S)$. We consider the case that exact computation of $S^*_{\boldsymbol{\mu}}$ may be computationally hard, and the algorithm may be randomized with a small failure probability. Thus, we resolve to the following (α, β)-*approximation oracle*:

- (α, β)**-Approximation oracle.** There is an (α, β)-approximation oracle for some $\alpha, \beta \le 1$ that takes an expectation vector $\boldsymbol{\mu}$ as input, and outputs a super arm $S \in \mathcal{S}$, such that $\Pr[r_{\boldsymbol{\mu}}(S) \ge \alpha \cdot \text{opt}_{\boldsymbol{\mu}}] \ge \beta$. Here β is the success probability of the oracle.

A lot of computationally hard problems do admit efficient approximation oracles (Vazirani , 2004). With an (α, β)-approximation oracle, it is no longer fair to compare the performance of a CMAB algorithm against the optimal reward $\text{opt}_{\boldsymbol{\mu}}$ as the regret of the algorithm. Instead, we compare against the $\alpha \cdot \beta$ fraction of the optimal reward, because only a β fraction of oracle computations are successful, and when successful the reward is only α-approximate of the optimal value. Formally, we define (α, β)-*approximation regret* of a CMAB algorithm A after n rounds of play using an (α, β)-approximation oracle under the expectation vector $\boldsymbol{\mu}$ as

$$Reg^A_{\boldsymbol{\mu},\alpha,\beta}(n) = n \cdot \alpha \cdot \beta \cdot \text{opt}_{\boldsymbol{\mu}} - \mathbb{E}_S \left[\sum_{t=1}^{n} r_{\boldsymbol{\mu}}(S_t^A) \right].$$

1: For each arm i, maintain: (1) variable T_i as the total number of times arm i is played so far; (2) variable $\hat{\mu}_i$ as the mean of all outcomes $X_{i,*}$'s of arm i observed so far.
2: For each arm i, play an arbitrary super arm $S \in \mathcal{S}$ such that $i \in S$ and update variables T_i and $\hat{\mu}_i$.
3: $t \leftarrow m$.
4: **while true do**
5: $\quad t \leftarrow t + 1$.
6: $\quad$ For each arm i, set $\bar{\mu}_i = \hat{\mu}_i + \sqrt{\frac{3 \ln t}{2T_i}}$.
7: $\quad S = \text{Oracle}(\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_m)$.
8: $\quad$ Play S and update all T_i's and $\hat{\mu}_i$'s.
9: **end while**

Algorithm 1: CUCB with computation oracle

Note that the classical MAB problem is a special case of our general CMAB problem, in which (a) the constraint $\mathcal{S} = [m]$ so that each super arm is just a simple arm; (b) the reward of a super arm $S = i$ in its t's trial is its outcome $X_{i,t}$; (c) the monotonicity and bounded smoothness hold trivially with function $f(\cdot)$ being the identity function; and (d) the (α, β)-approximation oracle is simply the argmax function among all expectation vectors, with $\alpha = \beta = 1$.

3. CUCB Algorithm for CMAB

We present our CUCB algorithm in Algorithm 1. After m initialization rounds, based on previous information, we maintain an empirical mean $\hat{\mu}_i$ for each arm i. More precisely, if arm i has been played s times by the end of round n, then the value of $\hat{\mu}_i$ at the end of round n is $(\sum_{j=1}^{s} X_{i,j})/s$. The actual expectation vector $\bar{\boldsymbol{\mu}}$ given to the oracle contains an adjustment term for each $\hat{\mu}_i$, which depends on the round number t and the number of times arm i has been played (stored in variable T_i). Then we simply play the super arm returned by the oracle and update variables T_i's and $\hat{\mu}_i$'s accordingly. Note that in our model all arms have bounded support on $[0, 1]$, but with the adjustment $\bar{\mu}_i$ may exceed 1. If such $\bar{\mu}_i$ is illegal to the oracle, we simply replace it with 1. Since replacing any value larger than 1 with 1 does not violate monotonicity and bounded smoothness of the reward function, our analysis below is not affected by this artifact and we directly use the original $\bar{\mu}_i$.

A super arm S is *bad* if $r_{\boldsymbol{\mu}}(S) < \alpha \cdot \text{opt}_{\boldsymbol{\mu}}$. We define $\mathcal{S}_{\text{B}} = \{S \mid r_{\boldsymbol{\mu}}(S) < \alpha \cdot \text{opt}_{\boldsymbol{\mu}}\}$ as the set of *bad* super arms. For a given underlying arm $i \in [m]$, we define

$$\Delta^i_{\min} = \alpha \cdot \text{opt}_{\boldsymbol{\mu}} - \max\{r_{\boldsymbol{\mu}}(S) \mid S \in \mathcal{S}_{\text{B}}, i \in S\}, \quad (1)$$

$$\Delta^i_{\max} = \alpha \cdot \text{opt}_{\boldsymbol{\mu}} - \min\{r_{\boldsymbol{\mu}}(S) \mid S \in \mathcal{S}_{\text{B}}, i \in S\}. \quad (2)$$

Furthermore, define $\Delta_{\max} = \max_{i \in [m]} \Delta^i_{\max}$ and $\Delta_{\min} = \min_{i \in [m]} \Delta^i_{\min}$. Our main theorem below provides the regret bound of the CUCB algorithm.

Theorem 1. *The (α, β)-approximation regret of the CUCB algorithm in n rounds using an (α, β)-approximation oracle is at most*

$$\sum_{i \in [m], \Delta^i_{\min} > 0} \left(\frac{6 \ln n \cdot \Delta^i_{\min}}{(f^{-1}(\Delta^i_{\min}))^2} + \int_{\Delta^i_{\min}}^{\Delta^i_{\max}} \frac{6 \ln n}{(f^{-1}(x))^2} \mathrm{d}x \right) + \left(\frac{\pi^2}{3} + 1 \right) \cdot m \cdot \Delta_{\max}. \quad (3)$$

where $f(\cdot)$ is the bounded smoothness function.

Due to the space constraint, we prove the following regret bound simplified from Eq.(3).

$$\left(\frac{6 \ln n}{\left(f^{-1}\left(\Delta_{\min}\right)\right)^2} + \frac{\pi^2}{3} + 1 \right) \cdot m \cdot \Delta_{\max}. \quad (4)$$

Proof of regret bound in Eq.(4). We use $\mathbb{I}\{\cdot\}$ to denote the indicator function, and $\mathbb{I}\{\mathcal{E}\} = 1$ if event $\mathcal{E}$ is true, and 0 if $\mathcal{E}$ is false.

For variable T_i, let $T_{i,t}$ be the value of T_i at the end of round t, that is, $T_{i,t}$ is the number of times arm i is played in the first t rounds. For variable $\hat{\mu}_i$, let $\hat{\mu}_{i,s}$ be the value of $\hat{\mu}_i$ after arm i is played s times, that is, $\hat{\mu}_{i,s} = (\sum_{j=1}^{s} X_{i,j})/s$. Then, the value of variable $\hat{\mu}_i$ at the end of round t is $\hat{\mu}_{i,T_{i,t}}$. For variable $\bar{\mu}_i$, let $\bar{\mu}_{i,t}$ be the value of $\bar{\mu}_i$ at the end of round t. Let $\bar{\boldsymbol{\mu}}_t = (\bar{\mu}_{1,t}, \ldots, \bar{\mu}_{m,t})$ be the random vector fed to the oracle as the input in line 7 of Algorithm 1 at round t.

In the t-th round, let F_t be the event that the oracle fails to produce an α-approximate answer with respect to the input vector $\bar{\boldsymbol{\mu}}_t = (\bar{\mu}_{1,t}, \ldots, \bar{\mu}_{m,t})$. We have $\Pr[F_t] = \mathbb{E}[\mathbb{I}\{F_t\}] \leq 1 - \beta$.

We maintain counter N_i for each arm i after the m initialization rounds. Let $N_{i,t}$ be the value of N_i after the t-th round and $N_{i,m} = 1$. Note that $\sum_i N_{i,m} = m$. Counters $\{N_i\}_{i=1}^m$ are updated as follows.

For a round $t > m$, let S_t be the super arm selected in round t by the oracle (line 7 of Algorithm 1). Round t is *bad* if the oracle selects a *bad* super arm $S_t \in \mathcal{S}_{\text{B}}$. If round t is bad, let $i = \text{argmin}_{j \in S_t} N_{j,t-1}$. We increment N_i by one, i.e., $N_{i,t} = N_{i,t-1} + 1$. That is, we find the arm i with the smallest counter in S_t and increment its counter. If i is not unique, we pick an arbitrary arm with the smallest counter in S_t. On the other hand, if $S_t \notin \mathcal{S}_B$, no counter will be incremented.

By definition $N_{i,t} \leq T_{i,t}$. Notice that in every bad round, exactly one counter in $\{N_i\}_{i=1}^m$ is incremented,

so the total number of bad rounds in the first n rounds is less than or equal to $\sum_i N_{i,n}$.

Define $\ell_t = \frac{6\ln t}{(f^{-1}(\Delta_{\min}))^2}$. Consider a bad round t, $S_t \in \mathcal{S}_{\mathrm{B}}$ is selected and counter N_i of some arm $i \in S_t$ is updated. We have

$$\begin{aligned}
&\sum_{i=1}^{m} N_{i,n} - m\cdot(\ell_n+1) = \sum_{t=m+1}^{n} \mathbb{I}\{S_t \in \mathcal{S}_{\mathrm{B}}\} - m\ell_n \\
\le &\sum_{t=m+1}^{n}\sum_{i\in[m]} \mathbb{I}\{S_t \in \mathcal{S}_{\mathrm{B}}, N_{i,t} > N_{i,t-1}, N_{i,t-1} > \ell_n\} \\
\le &\sum_{t=m+1}^{n}\sum_{i\in[m]} \mathbb{I}\{S_t \in \mathcal{S}_{\mathrm{B}}, N_{i,t} > N_{i,t-1}, N_{i,t-1} > \ell_t\} \\
= &\sum_{t=m+1}^{n} \mathbb{I}\{S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, N_{i,t-1} > \ell_t\} \qquad (5) \\
\le &\sum_{t=m+1}^{n} (\mathbb{I}\{F_t\} + \mathbb{I}\{\neg F_t, S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, N_{i,t-1} > \ell_t\}) \\
\le &\sum_{t=m+1}^{n} (\mathbb{I}\{F_t\} + \mathbb{I}\{\neg F_t, S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, T_{i,t-1} > \ell_t\}).
\end{aligned}$$

Eq.(5) holds by our rule of updating the counters. We first claim that $\Pr[\{\neg F_t, S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, T_{i,t-1} > \ell_t\}] \le 2\cdot m \cdot t^{-2}$.

In fact, for any $i \in [m]$,

$$\begin{aligned}
&\Pr\left[\mid \hat{\mu}_{i,T_{i,t-1}} - \mu_i \mid \ge \sqrt{3\ln t/(2T_{i,t-1})}\right] \\
=&\sum_{s=1}^{t-1} \Pr\left[\{\mid \hat{\mu}_{i,s} - \mu_i \mid \ge \sqrt{3\ln t/(2s)}, T_{i,t-1} = s\}\right] \\
\le&\sum_{s=1}^{t-1} \Pr\left[\mid \hat{\mu}_{i,s} - \mu_i \mid \ge \sqrt{3\ln t/(2s)}\right] \\
\le& t\cdot 2e^{-3\ln t} = 2t^{-2}, \qquad (6)
\end{aligned}$$

where the last inequality is due to the Chernoff-Hoeffding bound. Define $\Lambda_{i,t} = \sqrt{\frac{3\ln t}{2T_{i,t-1}}}$ (a random variable since $T_{i,t-1}$ is a random variable), and event $E_t = \{\forall i \in [m], |\hat{\mu}_{i,T_{i,t-1}} - \mu_i| \le \Lambda_{i,t}\}$. By union bound on Eq.(6), $\Pr[\neg E_t] \le 2\cdot m\cdot t^{-2}$. According to line 6 of Algorithm 1, we have $\bar{\mu}_{i,t} - \hat{\mu}_{i,T_{i,t-1}} = \Lambda_{i,t}$. Thus $|\hat{\mu}_{i,T_{i,t-1}} - \mu_i| \le \Lambda_{i,t}$ implies that $\bar{\mu}_{i,t} \ge \mu_i$.

Let $\Lambda = \sqrt{\frac{3\ln t}{2\ell_t}}$, which is not a random variable. Define random variable $\Lambda_t = \max\{\Lambda_{i,t} \mid i \in S_t\}$. Then

$$E_t \Rightarrow \forall i \in S_t, |\bar{\mu}_{i,t} - \mu_i| \le 2\Lambda_t \qquad (7)$$

$$\{S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, T_{i,t-1} > \ell_t\} \Rightarrow \Lambda > \Lambda_t \qquad (8)$$

Let $\bar{\boldsymbol{\mu}}_t = (\bar{\mu}_{1,t}, \ldots, \bar{\mu}_{m,t})$ be the vector representing the adjusted expectation vector at round t. Then,

$$E_t \Rightarrow \bar{\boldsymbol{\mu}}_t \ge \boldsymbol{\mu}. \qquad (9)$$

If $\{E_t, \neg F_t, S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, T_{i,t-1} > \ell_t\}$ holds at time t, we have the following important derivation:

$$\begin{aligned}
&r_{\boldsymbol{\mu}}(S_t) + f(2\Lambda) > r_{\boldsymbol{\mu}}(S_t) + f(2\Lambda_t) \ge r_{\bar{\boldsymbol{\mu}}_t}(S_t) \\
\ge&\alpha\cdot \mathrm{opt}_{\bar{\boldsymbol{\mu}}_t} \ge \alpha \cdot r_{\bar{\boldsymbol{\mu}}_t}(S^*_{\boldsymbol{\mu}}) \ge \alpha\cdot r_{\boldsymbol{\mu}}(S^*_{\boldsymbol{\mu}}) = \alpha\cdot \mathrm{opt}_{\boldsymbol{\mu}}.
\end{aligned}$$

The first inequality above is due to the strict monotonicity of $f(\cdot)$ and Eq.(8); the second is due to the bounded smoothness property and Eq.(7); the third is because $\neg F_t$ implies that S_t is an α approximation w.r.t $\bar{\boldsymbol{\mu}}_t$; the fourth is by the definition of $\mathrm{opt}_{\bar{\boldsymbol{\mu}}_t}$, and the last inequality is due to the monotonicity of $r_{\boldsymbol{\mu}}(S)$ and Eq.(9). So we have

$$r_{\boldsymbol{\mu}}(S_t) + f(2\Lambda) > \alpha\cdot \mathrm{opt}_{\boldsymbol{\mu}}. \qquad (10)$$

Since $\ell_t = \frac{6\ln t}{(f^{-1}(\Delta_{\min}))^2}$, we have $f(2\Lambda) = \Delta_{\min}$. Therefore, Eq. (10) contradicts the definition of $\Delta_{\min}$ and the fact that $S_t \in \mathcal{S}_{\mathrm{B}}$. In other words,

$$\begin{aligned}
&\Pr\left[\{E_t, \neg F_t, S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, T_{i,t-1} > \ell_t\}\right] = 0 \Rightarrow \\
&\Pr\left[\{\neg F_t, S_t \in \mathcal{S}_{\mathrm{B}}, \forall i \in S_t, T_{i,t-1} > \ell_t\}\right] \\
&\le \Pr[\neg E_t] \le 2\cdot m\cdot t^{-2}.
\end{aligned}$$

The claim thus holds. We have,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{m} N_{i,n}\right] \le m(\ell_n+1) + (1-\beta)(n-m) + \sum_{t=1}^{n}\frac{2m}{t^2} \\
&\le \frac{6m\cdot\ln n}{\left(f^{-1}\left(\Delta_{\min}\right)\right)^2} + (\frac{\pi^2}{3}+1)\cdot m + (1-\beta)(n-m).
\end{aligned}$$

Notice that each time we hit a bad super arm at time t, we incur a regret at most $\Delta_{\max} \ge \alpha\cdot\mathrm{opt}_{\mu} - r_{\mu}(S_t)$. Then we obtain the regret bound of Eq.(4) as follows.

$$\begin{aligned}
&Reg^{A}_{\boldsymbol{\mu},\alpha,\beta}(n) \\
\le&n\alpha\beta\cdot\mathrm{opt}_{\boldsymbol{\mu}} - \left(n\alpha\cdot\mathrm{opt}_{\boldsymbol{\mu}} - \mathbb{E}\left[\sum_{i=1}^{m} N_{i,n}\right]\cdot\Delta_{\max}\right) \\
\le&\left(\frac{6m\ln n}{\left(f^{-1}\left(\Delta_{\min}\right)\right)^2} + (\frac{\pi^2}{3}+1)m + (1-\beta)(n-m)\right) \\
&\cdot\Delta_{\max} - (1-\beta)n\cdot\alpha\cdot\mathrm{opt}_{\boldsymbol{\mu}} \\
\le&\left(\frac{6\ln n}{\left(f^{-1}\left(\Delta_{\min}\right)\right)^2} + \frac{\pi^2}{3} + 1\right)\cdot m\cdot\Delta_{\max}. \qquad \square
\end{aligned}$$

We now briefly discuss the idea to prove Theorem 1. In the proof of Eq.(4), we essentially show that if all arms are sufficiently sampled with respect to $\Delta_{\min}$, the probability that we hit a bad super arm is small. On the other hand, in a bad round, if the underlying arms

are not sufficiently sampled with respect to $\Delta_{\min}$, we incur a regret of $\Delta_{\max}$. Notice that there is a discrepancy in the analysis, i.e., the sufficiency of sampling is defined on $\Delta_{\min}$ while the regret is counted as $\Delta_{\max}$. Theorem 1 is based on a more refined analysis that defines the sufficiency of the sampling of arm i separately for each bad super arm containing i, which avoids the over charge above.

Comparing to classical MAB. The classical MAB is a special instance of our CMAB framework in which each super arm is a simple arm, function $f(x) = x$, and $\alpha = \beta = 1$. Notice that $\Delta^i_{\max} = \Delta^i_{\min}$. Thus, by Theorem 1, the regret bound of the classical MAB is

$$\sum_{i\in[m],\Delta^i>0} \frac{6\ln n}{\Delta^i} + \left(\frac{\pi^2}{3}+1\right)\cdot m\cdot\Delta_{\max}, \qquad (11)$$

where $\Delta^i = \max_{j\in[m]}\mu_j - \mu_i$. Comparing with the regret bound in Theorem 1 of (Auer et al., 2002a), we have a better coefficient $\sum_{i\in[m],\Delta^i>0} 6/\Delta^i$ in the leading $\ln n$ term than the original $\sum_{i\in[m],\Delta^i>0} 8/\Delta^i$.[1] The improvement is due to a tighter analysis, and is the reason that we obtained improved regret bound over (Gai et al., 2012) for the linear reward CMAB.

Our tight analysis implies a distribution-independent regret for arbitrary distributions with support in $[0,1]$ on all arms, for a large class of problem instances with a polynomial bounded smoothness function $f(x) = \gamma x^\omega$ for $\gamma > 0$ and $0 < \omega \le 1$, as shown below.

Theorem 2. *Consider a CMAB problem with an (α,β)-approximation oracle. If the bounded smoothness function $f(x) = \gamma\cdot x^\omega$ for some $\gamma > 0$ and $\omega \in (0,1]$, the regret of CUCB is at most:*

$$\frac{2\gamma}{2-\omega}\cdot(6m\ln n)^{\omega/2}\cdot n^{1-\omega/2} + \left(\frac{\pi^2}{3}+1\right)\cdot m\cdot\Delta_{\max}.$$

Note that when $\omega = 1$, which covers all applications discussed in Section 4, in the simple arm setting, we obtain a distribution-independent bound of $O(\sqrt{mn\ln n})$, which matches the original UCB1 algorithm (Audibert & Bubeck, 2009) (up to a logarithmic factor). In the linear combinatorial bandit setting, i.e., semi-bandit with L_∞ assumption in (Audibert et al., 2011), our regret is $O(\sqrt{m^3 n\log n})$, which is a factor $\sqrt{m}$ off the optimal bound in the adversarial setting.

4. Applications

In this section, we describe three applications that fit our CMAB framework. Notice that, the probabilistic maximum coverage bandit and social influence maximization bandit are instances of the online submodular maximization problem, which can be addressed in the adversarial setting by (Streeter & Golovin, 2008).

4.1. Probabilistic maximum coverage bandit

The online advertisement placement application discussed in the introduction can be modeled by the bandit version of the probabilistic maximum coverage (PMC) problem. PMC has as input a weighted bipartite graph $G = (L, R, E)$ where each edge (u,v) has a probability $p(u,v)$, and it needs to find a set $S \subseteq L$ of size k that maximizes the expected number of activated nodes in R, where a node $v \in R$ can be activated by a node $u \in S$ with an independent probability of $p(u,v)$. In the advertisement placement scenario, L is the set of web pages, R is the set of users, and $p(u,v)$ is the probability that user v clicks the advertisement on page u. PMC problem is NP-hard, since when all edge probabilities are 1, it becomes the NP-hard Maximum Coverage problem. Using submodular set function maximization technique (Nemhauser et al., 1978), it can be easily shown that there exists a deterministic $(1-1/e)$ approximation algorithm for the PMC problem, which means that we have a $(1-1/e,1)$-approximation oracle for PMC.

The PMC bandit problem is that edge probabilities are unknown, and one repeatedly selects k targets in L in multiple rounds, observes all edge activations and adjusts target selection accordingly in order to maximize the total number of activated nodes over all rounds. We can formulate this problem as an instance in the CMAB framework. Each edge $(u,v) \in E$ represents an arm, and each play of the arm is a 0-1 Bernoulli random variable with parameter $p_{u,v}$. A super arm is the set of edges E_S adjacent to a set $S \subseteq L$ of size k. The reward of E_S is the number of activated nodes in R, which is the number of nodes in R that are incident to at least one edge in E_S with outcome 1. Note that this reward is not linear to the outcomes of arms. The monotonicity property is straightforward. The bounded smoothness function is $f(x) = |E|\cdot x$, i.e., increasing all probabilities of all arms in a super arm by x can increase the expected number of activated nodes in V by at most $|E|\cdot x$. Since $f(\cdot)$ is a linear function, the integral in Eq.(3) has a closed form. In particular, by Theorem 1, we know that the $(1-1/e,1)$-approximation regret of our CUCB algorithm on PMC bandit is bounded by

$$\sum_{i\in E,\Delta^i_{\min}>0} \frac{12\cdot|E|^2\cdot\ln n}{\Delta^i_{\min}} + \left(\frac{\pi^2}{3}+1\right)\cdot|E|\cdot\Delta_{\max}.$$

Notice that all edges incident to a node $u \in L$ are

[1] We remark that the constant of UCB1 regret has been tightened to the optimum (Garivier & Cappé, 2011).

always played together. In this case, for any two edges $i, j \in E$ that are incident to the same node $u \in L$, we have $\Delta^i_{\min} = \Delta^j_{\min}$, and we define it to be $\Delta^u_{\min}$. The coefficient of $\ln n$ above could be written as $\sum_{u \in L, \Delta^u_{\min} > 0} 12 d_u |E|^2 / \Delta^u_{\min}$, where d_u is the degree of u. We call those arms that are always played together as *clustered arms*. In the supplementary material, we show how to exploit the arm clustering property to remove the d_u above and obtain the following better bound:

$$\sum_{u \in L, \Delta^u_{\min} > 0} \frac{12 \cdot |E|^2 \cdot \ln n}{\Delta^u_{\min}} + \left(\frac{\pi^2}{3} + 1\right) \cdot |E| \cdot \Delta_{\max}.$$

4.2. Social influence maximization bandit

In social influence maximization (Kempe et al., 2003), we are given a directed graph $G = (V, E)$, where every edge (u, v) is associated with an unknown *propagation probability* $p_{u,v}$. Initially, a seed set $S \subseteq V$ are selected and activated. In each iteration of the diffusion process, each node u activated in the previous iteration has one chance of activating its inactive outgoing neighbor v with probability $p_{u,v}$. The reward of S after the diffusion process is the total number of activated nodes in the end. Influence maximization is to find a seed set S of at most k nodes that maximize the expected reward. Kempe et al. (Kempe et al., 2003) show that the problem is NP-hard and provide an algorithm with approximation ratio $1-1/e-\varepsilon$ with success probability $(1 - 1/|E|)$ for any $\varepsilon > 0$. This means that we have a $(1 - 1/e - \varepsilon, 1 - 1/|E|)$-approximation oracle.

In the CMAB framework, we do not know the activation probabilities of edges and want to learn them during repeated seed selections while maximizing overall reward. Similar to PMC, we can treat each edge as an arm, and a super arm is a set of outgoing edges from at most k nodes. Different from PMC, when a super arm S is played, not only arms in S reveal their outcomes, but other arms (edges) may also reveal their outcomes in the diffusion process, and the reward depends on the outcomes of all these arms. As a result, our bounded smoothness function is $f(x) = |V| \cdot |E| \cdot x$. According to Theorem 1, the $(1-1/e-\varepsilon, 1-1/|E|)$-approximation regret of the CUCB algorithm on influence maximization is bounded by:

$$\sum_{i \in E, \Delta^i_{\min} > 0} \frac{12 \cdot |V|^2 \cdot |E|^2 \ln n}{\Delta^i_{\min}} + \left(\frac{\pi^2}{3} + 1\right) \cdot |E| \cdot \Delta_{\max}.$$

Similar to the PMC problem, we could exploit the arm clustering property and improve the regret bound.

4.3. Combinatorial bandits with linear rewards

Gai et al. (Gai et al., 2012) studied the *Learning with Linear Reward* policy (LLR). Their formulation is close to ours except that their reward function must be linear. In their setting, there are m underlying arms. Each super arm consists of a set of underlying arms S together with a set of coefficients $\{w_{i,S} \mid i \in S\}$. The reward of playing super arm S is $\sum_{i \in S} w_{i,S} \cdot X_i$, where X_i is the random outcome of arm i. The formulation can model a lot of bandit problems appeared in the literature, e.g., multiple plays, shortest path, minimum spanning tree and maximum weighted matching.

Our framework contains such linear reward problems as special cases.[2] In particular, let $L = \max_S |S|$ and $a_{\max} = \max_{i,S} w_{i,S}$, and we have the bounded smoothness function $f(x) = a_{\max} \cdot L \cdot x$. By applying Theorem 1, the regret bound is

$$\sum_{i \in [m], \Delta^i_{\min} > 0} \frac{12 \cdot a^2_{\max} \cdot L^2 \cdot \ln n}{\Delta^i_{\min}} + \left(\frac{\pi^2}{3} + 1\right) \cdot m \cdot \Delta_{\max}.$$

Our result significantly improves the coefficient of the leading $\ln n$ term comparing to Theorem 2 of (Gai et al., 2012) in two aspects: (a) we remove a factor of $L+1$; and (b) the coefficient $\sum_{i \in [m], \Delta^i_{\min} > 0} 1/\Delta^i_{\min}$ is likely to be much smaller than $m \cdot \Delta_{\max}/(\Delta_{\min})^2$ in (Gai et al., 2012). This demonstrates that while our framework covers a much larger class of problems, we are still able to provide much tighter analysis than the one for linear reward bandits.

5. Conclusion

In this paper, we propose the first general stochastic CMAB framework that accommodates a large class of nonlinear reward functions among combinatorial and stochastic arms. We provide CUCB algorithm with tight analysis on its distribution-dependent and distribution-independent regret bounds and applications to new practical combinatorial bandit problems.

There are many possible future directions from this work. One may study the CMAB problems with Markovian outcome distributions on arms, or the restless version of CMAB, in which the states of arms continue to evolve even if they are not played. Another direction is to apply CMAB to contextual bandit settings where arm distributions depend on the context of the play. One may also see if any technique of this work can be applied to the study of adversarial combinatorial bandits with nonlinear rewards. Of course, an important future work is to empirically validate our algorithm and demonstrate its effectiveness in practice.

[2] To include the linear reward case, we allow two super arms with the same set of underlying arms to have different sets of coefficients. This is fine as long as the oracle could output super arms with appropriate parameters.

References

Anantharam, V., Varaiya, P., and Walrand, J. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays — Part I: i.i.d. rewards; Part II: Markovian rewards. *IEEE Transactions on Automatic Control*, AC-32(11):968–982, 1987a.

Audibert, J.-Y., and Bubeck, S. Minimax policies for adversarial and stochastic bandits. In *COLT*, 2009.

Audibert, J.-Y., Bubeck, S., and Lugosi, G. Minimax policies for combinatorial prediction games. In *COLT*, 2011.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b.

Berry, D. and Fristedt, B. *Bandit problems.* Chapman and Hall, 1985.

Bubeck, S., Cesa-Bianchi, N., and Kakade, S. M. Towards Minimax Policies for Online Linear Optimization with Bandit Feedback. In *COLT*, 2012.

Caro, F. and Gallien, J. Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, 53:276–292, 2007.

Cesa-Bianchi, N. and Lugosi, G. Combinatorial bandits.

Gai, Y., Krishnamachari, B., and Jain, R. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *DySPAN*, 2010.

Gai, Y., Krishnamachari, B., and Jain, R. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20, 2012.

Garivier, A. and Cappé, O. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *COLT*, 2011.

Hazan, E. and Kale, S. Online submodular minimization. In *NIPS*, 2009.

Kakade, S. M., Kalai, A. T., and Ligett, K. Playing games with approximation algorithms. *SIAM Journal on Computing*, 39(3):1088–1106, 2009.

Kempe, D., Kleinberg, J. M., and Tardos, É. Maximizing the spread of influence through a social network. In *KDD*, 2003.

Lai, T. L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

Liu, H., Liu, K., and Zhao, Q. Logarithmic weak regret of non-bayesian restless multi-armed bandit. In *ICASSP*, 2011.

Liu, K. and Zhao, Q. Adaptive shortest-path routing under unknown and stochastically varying link states. *Arxiv preprint arXiv:1201.4906*, 2012.

Mannor, S., and Shamir, O. From Bandits to Experts: On the Value of Side-Observations. In *NIPS*, 2011.

Nemhauser, G., Wolsey, L., and Fisher, M. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

Radlinski, F., Kleinberg, R., and Joachims, T. Learning diverse rankings with multi-armed bandits. In *ICML*, 2008.

Streeter, M., Golovin, D., and Krause, A. Online learning of assignments. In *NIPS*, 2009.

Streeter, M. and Golovin, D. An online algorithm for maximizing submodular functions. In *NIPS*, 2008.

Sutton, R. and Barto, A. *Reinforcement learning, an introduction.* MIT Press, 1998.

Vazirani, V. V. *Approximation Algorithms* Springer, 2004