Thompson Sampling for Combinatorial Semi-Bandits

Siwei Wang [1] **Wei Chen** [2]

Abstract

We study the application of the Thompson sampling (TS) methodology to the stochastic combinatorial multi-armed bandit (CMAB) framework. We analyze the standard TS algorithm for the general CMAB, and obtain the first distribution-dependent regret bound of $O(m \log T/\Delta_{\min})$ for TS under general CMAB, where m is the number of arms, T is the time horizon, and $\Delta_{\min}$ is the minimum gap between the expected reward of the optimal solution and any non-optimal solution. We also show that one cannot use an approximate oracle in TS algorithm for even MAB problems. Then we expand the analysis to matroid bandit, a special case of CMAB and for which we could remove the independence assumption across arms and achieve a better regret bound. Finally, we use some experiments to show the comparison of regrets of CUCB and CTS algorithms.

1. Introduction

Multi-armed bandit (MAB) (Berry & Fristedt, 1985; Sutton & Barto, 1998) is a classical online learning model typically described as a game between a learning agent (player) and the environment with m arms. In each step, the environment generates an outcome, and the player uses a policy (or an algorithm), which takes the feedback from the previous steps as input, to select an arm to pull. After pulling an arm, the player receives a reward based on the pulled arm and the environment outcome. In this paper, we consider stochastic MAB problem, which means the environment outcome is drawn from an unknown distribution (Lai & Robbins, 1985), not generated by an adversary (Auer et al., 2002b). The goal of the player is to cumulate as much reward as possible over a total of T steps (T may be unknown). The performance metric is the *(expected) regret*, which is the cumulative difference over T steps between always playing the arm with the optimal expected reward and playing the arms according to the policy.

MAB models the key tradeoff between exploration — continuing exploring new arms not observed often, and exploitation — sticking to the best performed arm based on the observation so far. A famous MAB algorithm is the upper confidence bound (UCB) policy (Gittins, 1989; Auer et al., 2002a), which achieves $O(m \log T/\Delta)$ distribution-dependent regret, where Δ is the minimum gap in the expected reward between an optimal arm and any non-optimal arm, and it matches the lower bound (Lai & Robbins, 1985).

Combinatorial multi-armed bandit (CMAB) problem has recently become an active research area (Gai et al., 2012; Chen et al., 2016b; Gopalan et al., 2014a; Kveton et al., 2014; 2015a;b; Wen et al., 2015; Combes et al., 2015; Chen et al., 2016a; Wang & Chen, 2017). In CMAB, the environment contains m *base arms*, but the player needs to pull a set of base arms S in each time slot, where S is called a *super arm* (or an *action*). The kind of reward and feedback varies in different settings. In this paper, we consider the semi-bandit setting, where the feedback includes the outcomes of all base arms in the played super arm, and the reward is a function of S and the observed outcomes of arms in S. CMAB has found applications in many areas such as wireless networking, social networks, online advertising, etc. Thus it is important to investigate different approaches to solve CMAB problems.

An alternative approach different from UCB is the Thompson sampling (TS) approach, which is introduced much earlier by Thompson (1933), but the theoretical analysis of the TS policy comes much later — Kaufmann et al. (2012) and Agrawal & Goyal (2012) give the first regret bound for the TS policy, which essentially matches the UCB policy theoretically. Moreover, TS policy often performs better than UCB in empirical simulation results, making TS an attractive policy for further studies.

TS policy follows the Bayesian inference framework to solve the MAB problems. The unknown distribution of environment outcomes is parameterized, with an assumed prior distribution. TS updates the prior distribution in each step with two phases: first it uses the prior distribution to sample a parameter, which is used to determine the action to

[1] Tsinghua University, Beijing, China [2] Microsoft Research, Beijing, China. Correspondence to: Siwei Wang <wangsw15@mails.tsinghua.edu.cn>, Wei Chen <weic@microsoft.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

play in the current step; second it uses the feedback obtained in the current step to update the prior distribution to posterior distribution according to the Bayes' rule. To avoid confusion on these two kinds of random variables, in the rest of this paper, we use the word "sample" to denote the variable in the first phase, i.e. the random variable coming from the prior distribution. The word "observation" represents the feedback random variable, which follows the unknown environment distribution.

In this paper, we study the application of the Thompson sampling approach to CMAB. The reason that we are interested in this approach is that it has good performance in experiments. We found out that TS-based policy performs better than many kinds of UCB-based policy in experiments. We also adjust the parameters of UCB-based policy to make it behave better, but those parameters do not have theoretical guarantees. Thompson sampling policy behaves almost the same as UCB-based policies with no-guarantee parameters, and much better than those with parameters that have theoretical regret bounds. We can see those results in our experiments from Section 5. Another interesting thing is that TS-based policy only require the reward function to be continuous, while UCB-based policy need it to be monotone as well. These make TS-based policy more competitive in real applications.

We consider a general CMAB case similar with (Chen et al., 2016b), i.e. we assume that (a) the problem instance satisfies a Lipschitz continuity assumption to handle non-linear reward functions, and (b) the player has access to an exact oracle for the offline optimization problem. We use the standard TS policy together with the offline oracle, and refer it as the combinatorial Thompson sampling (CTS) algorithm.

CTS policy would first derive a set of parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_m)$ as sample set for the base arms, and then select the optimal super arm under $\boldsymbol{\theta}$. The original analysis for TS on MAB model then faces a challenge in addressing the dependency issue: it essentially requires that different super arms be related with independent samples so that when comparing them and selecting the optimal super arm, the actual optimal one is selected with high probability. But when super arms are based on the *same* sample set $\boldsymbol{\theta}$, dependency and correlation among super arms may likely fail the above high probability analysis.

One way to get around this is to independently derive a sample set $\boldsymbol{\theta}(S)$ for every super arm S and compute its expected reward under $\boldsymbol{\theta}(S)$, and then select the optimal super arm. Obviously this solution incurs exponential sampling cost and is what we want to avoid when solving CMAB.

To address the dependency challenge, we adapt an analysis of (Komiyama et al., 2015) for selecting the top-k arms to the general CMAB setting. The adaptation is nontrivial since we are dealing with arbitrary combinatorial constraints while they only deal with super arms containing k arms.

We show that CTS achieves $O(\sum_{i\in[m]} \log T/\Delta_{i,\min}) + O((2/\varepsilon)^{2k^*})$ distribution-dependent regret bound for some small ε, where $\Delta_{i,\min}$ is the minimum gap between the optimal expected reward and any non-optimal expected reward containing arm i, and k^* is the size of the optimal solution. This is the first distribution-dependent regret bound for general CMAB using TS-based policy, and the result matches the theoretical performance of the UCB-based solution CUCB in (Chen et al., 2016b). When considering CMAB with linear reward functions, the other complexity factors in the leading $\log T$ term also matches the regret lower bound for linear CMAB in (Kveton et al., 2015a). For the exponential constant term, we show an example that it is unavoidable for Thompson sampling.

Comparing to the UCB-based solution in (Chen et al., 2016b), the advantages of CTS is that: a) we do not need to assume that the expected reward is monotone to the mean outcomes of the base arms; b) it has better behaviour in experiments. CTS also suffers from some disadvantages. For example, CTS policy can not adapt an approximation oracle as in (Chen et al., 2016b) (the regret becomes to approximate regret as well). However, we claim that it is because of the difference between TS-based algorithm and UCB-based algorithm. To show this, we provide a counter example for origin MAB problem, which cause an approximate regret of $\Theta(T)$ when using TS policy.

Another disadvantage is that we need to assume that all the outcomes of all base arms are mutually independent. This is because TS policy maintains a prior distribution for every base arm's mean value μ_i. Only when the distributions are independent, we can use a simple method to update those prior distributions; otherwise the update method will be much more complicated for both the implmentation and the analysis. This assumption is still reasonable, since many real applications satisfy this assumption.

However, when applying on some further combinatorial structures, we do not need such an assumption, such as in *matroid bandit*. Matroid bandit is a special class of CMAB (Kveton et al., 2014), in which the base arms are the elements in the ground set and the super arms are the independent sets of a matroid. The reward function is the sum of all outcomes of the base arms in the super arm. We show that the regret of CTS is upper bounded by $O(\sum_{i\notin S^*} \log T/\Delta_i) + O(m/\varepsilon^4)$ for some small ε, where S^* is an optimal solution and Δ_i is the minimum positive gap between the mean outcome of any arms in S^* and the mean outcome of arm i. This result does not need to assume that all arm distributions are independent, and do not have a constant term exponential with k^*. It matches both the theoretical performance of the UCB-based algorithm and the lower bound given in Kveton et al. (2014), and the constant term is similar with results in Agrawal & Goyal (2012), which appears in almost every TS analysis paper.

We further conduct empirical simulations, and show that CTS performs much better than the CUCB algorithm of (Chen et al., 2016b) and C-KL-UCB algorithm based on KL-UCB of (Garivier & Cappé, 2011) on both matroid and non-matroid CMAB problem instances.

In summary, our contributions include that: (a) we provide a novel analysis for the general CMAB problem, and provides the first distribution-dependent regret bound for general CMAB problems and matroid bandit problems based on Thompson sampling; (b) we show that approximation oracle can not be used in TS-based algorithms; and (c) we show that the exponential constant in our regret bound for general CMAB problems is unavoidable.

Due to space constraint, complete proofs are moved to the supplementary material.

1.1. Related Work

A number of related works on the general context of multi-armed bandit and Thompson sampling have been given, and we focus here on the most relevant studies related to CMAB.

Our study follows the general CMAB framework of (Chen et al., 2016b), which provides a UCB-style algorithm CUCB and show a $O(\sum_{i\in[m]}\log T/\Delta_{i,\min})$ regret bound. Comparing with our CTS, both use an offline oracle, assume a Lipschitz continuity condition, and the bound essentially match asymptotically on time horizon T. Their differences include: (a) CTS does not need the monotonicity property but CUCB requires that to use the offline oracle; (b) CUCB allows an approximation oracle (and uses approximate regret), but CTS requires an exact oracle; (c) the regret bound of CTS has some additional terms not related to T, which is common in TS-based regret bounds (See Section 3 for more details). Combes et al. (2015) propose ESCB algorithm to solve CMAB with linear reward functions and independent arms, and their regret is a factor $O(\sqrt{m})$ better than our corresponding regret bound for CTS. However, their ESCB algorithm requires an exponential-time for computation, which is what we want to avoid when designing CMAB algorithms.

Matroid bandit is defined and studied by Kveton et al. (2014), who provide a UCB-based algorithm with regret bound almost exactly matches CTS algorithm. They also prove a matching lower bound using a partition matroid bandit.

Thompson sampling has also been applied to settings with combinatorial actions. Gopalan et al. (2014b) study a general action space with a general feedback model, and provide analytical regret bounds for the exact TS policy. However, their general model cannot be applied to our case. In particular, they assume that the arm outcome distribution is from a known parametric family and the prior distribution on the parameter space is finitely supported. We instead work on arbitrary nonparametric and unknown distributions with bounded support, and even if we work on a parametric family, we allow the support of prior distributions to be infinite or continuous. The reason is that in our CMAB setting, we only need to learn the means of base arms (same as in (Chen et al., 2016b)). Moreover, their regret bounds are high probability bounds, not expected regret bounds, and their bounds contain a potentially very large constant, which will turn into a non-constant term when we convert them to expected regret bounds.

In (Komiyama et al., 2015), the authors consider the TS-based policy for the top-k CMAB problem, a special case of matroid bandits where the super arms are subsets of size at most k. Thus we generalize top-k bandits to matroid bandits, and our regret bound for matroid bandit still matches the one in (Komiyama et al., 2015).

Wen et al. (2015) analyze the regret of using TS policy for contextual CMAB problems. The key difference between their work and ours is that they use the Bayesian regret metric. Bayesian regret takes another expectation on the prior distribution of parameters, while our regret bound works for *any* given parameter. This leads to very different analytical method, and they cannot provide distribution-dependent regret bounds. Russo & Van Roy (2016) also use Bayesian regret to analyze the regret bounds of TS policy for any kind of MAB problems. Again, due to the use of Bayesian regret, their analytical method is very different and cannot be used for our purpose.

2. Model and Definitions

2.1. CMAB Problem Formulation

A CMAB problem instance is modeled as a tuple $([m],\mathcal{I},D,R,Q)$. $[m]=\{1,2,\cdots,m\}$ is the set of base arms; $\mathcal{I}\subseteq 2^{[m]}$ is the set of super arms; D is a probability distribution in $[0,1]^m$, and is unknown to the player, R and Q are reward and feedback functions to be specified shortly. Let $\boldsymbol{\mu}=(\mu_1,\cdots,\mu_m)$, where $\mu_i=\mathbb{E}_{\boldsymbol{X}\sim D}[X_i]$. At discrete time slot $t\geq 1$, the player pulls a super arm $S(t)\in\mathcal{I}$, and the environment draws a random outcome vector $\boldsymbol{X}(t)=\{X_1(t),\cdots,X_m(t)\}\in[0,1]^m$ from D, independent of any other random variables. Then the player receives an unknown reward $R(t)=R(S(t),\boldsymbol{X}(t))$, and observes the feedback $Q(t)=Q(S(t),\boldsymbol{X}(t))$. As in (Chen et al., 2016b) and other papers studying CMAB, we consider semi-bandit feedback, that is, $Q(t)=\{(i,X_i(t))\mid i\in S(t)\}$. At time t, the previous step information is

$\mathcal{F}_{t-1} = \{(S(\tau), Q(\tau)) : 1 \leq \tau \leq t-1\}$, which is the input to the learning algorithm to select the action $S(t)$. Similar to (Chen et al., 2016b), we make the following two assumptions. For a parameter vector $\boldsymbol{\mu}$, we use $\boldsymbol{\mu}_S$ to denote the projection of $\boldsymbol{\mu}$ on S, where S is a subset of all the base arms.

Assumption 1. *The expected reward of a super arm $S \in \mathcal{I}$ only depends on the mean outcomes of base arms in S. That is, there exists a function r such that $\mathbb{E}[R(t)] = \mathbb{E}_{\boldsymbol{X}(t)\sim D}[R(S(t), \boldsymbol{X}(t))] = r(S(t), \boldsymbol{\mu}_{S(t)})$.*

The second assumption is a Lipschitz-continuity assumption of function r to deal with non-linear reward functions (it is based on one-norm).

Assumption 2. *There exists a constant B, such that for every super arm S and every pair of mean vectors $\boldsymbol{\mu}$ and $\boldsymbol{\mu'}$, $|r(S,\boldsymbol{\mu}) - r(S,\boldsymbol{\mu'})| \leq B||\boldsymbol{\mu}_S - \boldsymbol{\mu'}_S||_1$.*

The goal of the player is to minimize the total (expected) regret under time horizon T, as defined below:

$$Reg(T) \triangleq \mathbb{E}\left[\sum_{t=1}^{T}(r(S^*,\boldsymbol{\mu}) - r(S(t),\boldsymbol{\mu}))\right],$$

where $S^* \in \operatorname{argmax}_{S\in\mathcal{I}} r(S,\boldsymbol{\mu})$ is a best super arm.

2.2. Matroid Bandit

In matroid bandit settings, $([m], \mathcal{I})$ is a matroid, which means that $\mathcal{I}$ has two properties:

- If $A \in \mathcal{I}$, then $\forall A' \subseteq A$, $A' \in \mathcal{I}$;
- If $A_1, A_2 \in \mathcal{I}$, $|A_1| > |A_2|$, then there exists $i \in A_1 \setminus A_2$ such that $A_2 \cup \{i\} \in \mathcal{I}$.

The reward function is $R(S, \boldsymbol{x}) = \sum_{i\in S} x_i$, and thus the expected reward function is $r(S,\boldsymbol{\mu}) = \sum_{i\in S}\mu_i$.

3. Combinatorial Thompson Sampling

We first consider the general CMAB setting. For this setting, we assume that the player has an exact oracle $\mathsf{Oracle}(\boldsymbol{\theta})$ that takes a vector of parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)$ as input, and output a super arm $S = \operatorname{argmax}_{S\in\mathcal{I}} r(S,\boldsymbol{\theta})$.

The combinatorial Thompson sampling (CTS) algorithm is described in Algorithm 1. Initially we set the prior distribution of the means of all base arms as the Beta distribution $\beta(1,1)$, which is the uniform distribution on $[0,1]$. After we get observation $Q(t)$, we update the prior of all base arms in $S(t)$ using procedure Update (Algorithm 2): for each observation $X_i(t)$, we generate a Bernoulli random variable $Y_i(t)$ (the value of Y_i at time t) independently with mean $X_i(t)$, and then we update the prior Beta distribution of base arm i using $Y_i(t)$ as the new observation. It is easy to see that $\{Y_i(t)\}_t$'s are i.i.d. with the same mean μ_i as the samples $\{X_i(t)\}_t$'s. Let $a_i(t)$ and $b_i(t)$ denote the values of a_i and b_i at the beginning of time step t. Then, following the Bayes' rule, the posterior distribution of parameter μ_i after observation $Q(t)$ is $\beta(a_i(t)+Y_i(t), b_i(t)+1-Y_i(t))$, which is what the Update procedure does for updating a_i and b_i. When choosing a super arm, we simply draw independent samples from all base arms' prior distributions, i.e. $\theta_i(t) \sim \beta(a_i(t), b_i(t))$, and then send the sample vector $\boldsymbol{\theta}(t) = (\theta_1(t), \ldots, \theta_m(t))$ to the oracle. We use the output from the oracle $S(t)$ as the super arm to play.

Algorithm 1 CTS Algorithm for CMAB

```
For each arm $i$, let $a_i = b_i = 1$
for $t = 1, 2, \cdots$ do
    For all arm $i$, draw a sample $\theta_i(t)$ from Beta distribution $\beta(a_i, b_i)$; let $\boldsymbol{\theta}(t) = (\theta_1(t), \ldots, \theta_m(t))$
    Play action $S(t) = \mathsf{Oracle}(\boldsymbol{\theta}(t))$, get the observation $Q(t) = \{(i, X_i(t)) : i \in S(t)\}$
    $\mathsf{Update}(\{(a_i, b_i) \mid i \in S(t)\}, Q(t))$
end for
```

Algorithm 2 Procedure Update

```
Input: $\{(a_i, b_i) \mid i \in S\}$, $Q = \{(i, X_i) \mid i \in S\}$
Output: updated $\{(a_i, b_i) \mid i \in S\}$
for all $(i, X_i) \in Q$ do
    $Y_i \leftarrow 1$ with probability $X_i$, $0$ with probability $1-X_i$
    $a_i \leftarrow a_i + Y_i$; $b_i \leftarrow b_i + 1 - Y_i$
end for
```

We also need a further assumption to tackle the problem:

Assumption 3. *$D = D_1 \times D_2 \times \cdots \times D_m$, i.e., the outcomes of all base arms are mutually independent.*

This assumption is not necessary in CUCB algorithms. However, when using TS method, this assumption is needed. This is because that we are using the Bayes' Rule, thus we need the exact likelihood function (as we can see in (Gopalan et al., 2014b)). Only when the distributions for all the base arms are independent, we can use the Update procedure (Algorithm 2) to update their mean vector's prior distribution. When the distributions are correlated, the update procedure will also be much more complicated.

3.1. Regret Upper Bound

Let $\mathsf{OPT} = \operatorname{argmax}_{S\in\mathcal{I}} r(S,\boldsymbol{\mu})$ be the set of optimal super arms. Let $S^* \in \operatorname{argmin}_{S\in\mathsf{OPT}} |S|$ is one of the optimal super arm with minimum size k^*. Then we can define $\Delta_S = r(S^*,\boldsymbol{\mu}) - r(S,\boldsymbol{\mu})$, and $\Delta_{\max} = \max_{S\in\mathcal{I}} \Delta_S$. $K_{\max}$ is the maximum size of super arms, i.e. $K_{\max} = \max_{S\in\mathcal{I}} |S|$.

Theorem 1. *Under Assumptions 1, 2, and 3, for all D,* *Algorithm 1 has regret upper bound*

$$\sum_{i=1}^{m} \quad \max_{S:i\in S} \frac{8B^2|S|\log T}{\Delta_S - 2B(k^{*2}+2)\varepsilon} + (\frac{mK_{\max}^2}{\varepsilon^2} + 3m)\Delta_{\max} + \alpha_1 \cdot \left(\frac{8\Delta_{\max}}{\varepsilon^2}(\frac{4}{\varepsilon^2}+1)^{k^*}\log\frac{k^*}{\varepsilon^2}\right), \tag{1}$$

for any ε such that $\forall S, \Delta_S > 2B(k^{*2}+2)\varepsilon$, where B is the Lipschitz constant in Assumption 2, and α_1 is a constant not dependent on the problem instance.

When ε is sufficiently small, the leading $\log T$ term in the regret bound is comparable with the regret bound for CUCB in (Chen et al., 2016b). The term related to ε is to handle continuous Beta prior — since we will never be able to sample a $\theta_i^{(k)}(t)$ to be exactly the true value μ_i, we need to consider the ε neighborhood of μ_i. This ε term is common in most Thompson sampling analysis.

The constant term has an exponential dependency on k^*. This is because we need all the k^* base arms in the best super arm to have samples close to their means to make sure that it is the best super arm in sampling. In contrast, for the top-k MAB of (Komiyama et al., 2015), there is no such exponential dependency, because they only compare one base arm at a time (this can also be seen in our matroid Bandit analysis). When dealing with the general actions, the regret result of (Gopalan et al., 2014b) also contains an exponentially large constant term without a close form, which is likely to be much larger than ours. In Section 3.3, we show that this exponential constant is unavoidable for the general CTS.

We now provide the proof outline for Theorem 1. First we define the following four events:

- $\mathcal{A}(t) = \{S(t) \notin \mathsf{OPT}\}$
- $\mathcal{B}(t) = \{\exists i \in S(t), |\hat{\mu}_i(t) - \mu_i| > \frac{\varepsilon}{|S(t)|}\}$
- $\mathcal{C}(t) = \{||\boldsymbol{\theta}_{S(t)}(t) - \boldsymbol{\mu}_{S(t)}||_1 > \frac{\Delta_{S(t)}}{B} - (k^{*2}+1)\varepsilon\}$
- $\mathcal{D}(t) = \{\exists i \in S(t), |\theta_i(t) - \hat{\mu}_i(t)| > \sqrt{\frac{2\log T}{N_i(t)}}\}$

Then the total regret can be written as:

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}[\mathcal{A}(t)] \times \Delta_{S(t)}\right] \\
\le &\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}[\mathcal{B}(t) \wedge \mathcal{A}(t)] \times \Delta_{S(t)}\right] \\
+ &\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}[\neg\mathcal{B}(t) \wedge \mathcal{C}(t) \wedge \mathcal{D}(t) \wedge \mathcal{A}(t)] \times \Delta_{S(t)}\right] \\
+ &\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}[\neg\mathcal{B}(t) \wedge \mathcal{C}(t) \wedge \neg\mathcal{D}(t) \wedge \mathcal{A}(t)] \times \Delta_{S(t)}\right] \\
+ &\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{I}[\neg\mathcal{C}(t) \wedge \mathcal{A}(t)] \times \Delta_{S(t)}\right]
\end{aligned}$$

The first term can be bounded by Chernoff Bound, which is $\left(\frac{mK_{\max}^2}{\varepsilon^2} + m\right)\Delta_{\max}$. The second term can be bounded by some basic results of Beta distribution, the upper bound is $2m\Delta_{\max}$.

The third term is a little bit tricky. Notice that under $\mathcal{C}(t)$, we can use $B||\boldsymbol{\theta}_{S(t)}(t) - \boldsymbol{\mu}_{S(t)}||_1$ as an approximation of $\Delta_{S(t)}$. However, it is hard to bound $\sum_{i\in S(t)} |\theta_i(t) - \mu_i|$. To deal with this, we say one base arm $i \in S(t)$ is sufficiently learned if $N_i(t) > L_i(S(t)) = 2\log T/(\frac{\Delta_{S(t)}}{2B|S(t)|} - \frac{k^{*2}+2}{|S(t)|}\varepsilon)^2$. When computing $\sum_{i\in S(t)} |\theta_i(t) - \mu_i|$, we do not count all the sufficiently learned arms in. To compensate their contributions, we double all the insufficiently learned arms' contributions from $\sqrt{\frac{2\log T}{N_i(t)}}$ to $2\sqrt{\frac{2\log T}{N_i(t)}}$. One can check that the sum of contributions from insufficiently learned arms is an upper bound for the regret $\Delta_{S(t)}$ under $\neg\mathcal{B}(t) \wedge \mathcal{C}(t) \wedge \neg\mathcal{D}(t)$. Thus, we can upper bound the total contribution of base arm i as: $\sum_i 4B\sqrt{2\log T L_i^{\max}}$ where $L_i^{\max} = \max_{S:i\in S} L_i(S)$.

The difficulty comes mainly from the last term. Although the $\theta_i(t)$'s of all base arms are mutually independent, when it comes to super arms, the value $r(S, \boldsymbol{\theta}(t)_S)$'s for different super arms S are not mutually independent, because super arms may overlap one another. For example, Lemma 1 in (Agrawal & Goyal, 2013) is not true for considering super arms because of the lack of independence. This means that we cannot simply use the technique of (Agrawal & Goyal, 2013). Dealing with the dependency issue for this case is the main novelty in our analysis, as we now explain.

Let $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_m)$ be a vector of parameters, $Z \subseteq [m]$ and $Z \neq \emptyset$ be some base arm set and Z^c be the complement of Z. Recall that $\boldsymbol{\theta}_Z$ is the sub-vector of $\boldsymbol{\theta}$ projected onto Z, and we use notation $(\boldsymbol{\theta}'_Z, \boldsymbol{\theta}_{Z^c})$ to denote replacing θ_i's with θ'_i's for $i \in Z$ and keeping the values θ_i for $i \in Z^c$ unchanged.

Given a subset $Z \subseteq S^*$, we consider the following property for $\boldsymbol{\theta}_{Z^c}$. For any $||\boldsymbol{\theta}'_Z - \boldsymbol{\mu}_Z||_\infty \le \varepsilon$, let $\boldsymbol{\theta}' = (\boldsymbol{\theta}'_Z, \boldsymbol{\theta}_{Z^c})$:

- $Z \subseteq \mathsf{Oracle}(\boldsymbol{\theta}')$;
- Either $\mathsf{Oracle}(\boldsymbol{\theta}') \in \mathsf{OPT}$ or $||\boldsymbol{\theta}'_{\mathsf{Oracle}(\boldsymbol{\theta}')} - \boldsymbol{\mu}_{\mathsf{Oracle}(\boldsymbol{\theta}')}||_1 > \frac{\Delta_{\mathsf{Oracle}(\boldsymbol{\theta}')}}{B} - (k^{*2}+1)\varepsilon$.

The first one is to make sure that if we have normal samples in Z at time t, then arms in Z will be played and observed.

These observations would update the Beta distributions of these arms to be more accurate, such that it is easier next time that the samples from these arms are also within ε of their true value. This fact would be used later in the quantitative regret analysis. The second one says that if the samples in Z are normal, then $\neg\mathcal{C}(t) \wedge \mathcal{A}(t)$ can not happen (similar to the analysis in (Agrawal & Goyal, 2013) and (Komiyama et al., 2015)). As time going on, the probability that $\neg\mathcal{C}(t) \wedge \mathcal{A}(t)$ happens will become smaller and smaller, thus the expectation on its sum has an constant upper bound.

We use $\mathcal{E}_{Z,1}(\boldsymbol{\theta})$ to denote the event that the vector $\boldsymbol{\theta}_{Z^c}$ has such a property, and emphasize that this event only depends on the values in vector $\boldsymbol{\theta}_{Z^c}$. What we want to do is to find some exact Z such that $\mathcal{E}_{Z,1}(\boldsymbol{\theta})$ happens when $\neg\mathcal{C}(t) \wedge \mathcal{A}(t)$ happens. The following lemma shows that such Z must exist, it is the key lemma in this section.

Lemma 1. *Suppose that $\neg\mathcal{C}(t) \wedge \mathcal{A}(t)$ happens, then there exists $Z \subseteq S^*$ and $Z \neq \emptyset$ such that $\mathcal{E}_{Z,1}(\boldsymbol{\theta}(t))$ holds.*

By Lemma 1, for some nonempty Z, $\mathcal{E}_{Z,1}(\boldsymbol{\theta}(t))$ occurs when $\neg\mathcal{C}(t) \wedge \mathcal{A}(t)$ happens. Another fact is that $||\boldsymbol{\theta}_Z(t) - \boldsymbol{\mu}_Z||_\infty > \varepsilon$. The reason is that if $||\boldsymbol{\theta}_Z(t) - \boldsymbol{\mu}_Z||_\infty \le \varepsilon$, by definition of the property, either $S(t) \in \mathsf{OPT}$ or $||\boldsymbol{\theta}_{S(t)}(t) - \boldsymbol{\mu}_{S(t)}||_1 > \frac{\Delta_{S(t)}}{B} - ({k^*}^2 + 1)\varepsilon$, which means $\neg\mathcal{C}(t) \wedge \mathcal{A}(t)$ can not happen. Let $\mathcal{E}_{Z,2}(\boldsymbol{\theta})$ be the event $\{||\boldsymbol{\theta}_Z - \boldsymbol{\mu}_Z||_\infty > \varepsilon\}$. Then $\{\neg\mathcal{C}(t) \wedge \mathcal{A}(t)\} \rightarrow \vee_{Z \subseteq S^*, Z \neq \emptyset}(\mathcal{E}_{Z,1}(\boldsymbol{\theta}(t)) \wedge \mathcal{E}_{Z,2}(\boldsymbol{\theta}(t)))$.

Using similar techniques in (Komiyama et al., 2015) we can get the upper bound $O\left(\frac{8}{\varepsilon^2}(\frac{4}{\varepsilon^2})^{|Z|}\log\frac{|Z|}{\varepsilon^2}\right)$ for $\sum_{t=1}^T \mathbb{E}\left[\mathbb{I}\{\mathcal{E}_{Z,1}(\boldsymbol{\theta}(t)), \mathcal{E}_{Z,2}(\boldsymbol{\theta}(t))\}\right]$.

3.2. Approximation Oracle

We consider using an approximation oracle in our CTS algorithm as well, like what the author did in (Chen et al., 2016b) or (Wen et al., 2015). However, we found out that Thompson sampling does not work with an approximation oracle even in the original MAB model, as shown in Theorem 2. Notice that here we do not consider the Bayesian regret, so it does not contradict with the results in (Wen et al., 2015).

To make it clear, we need to show the definitions of approximation oracle and approximation regret here.

Definition 1. *An approximation oracle with rate λ for MAB problem is a function* $\mathsf{Oracle} : [0,1]^m \rightarrow \{1, \cdots, m\}$ *such that* $\mu_{\mathsf{Oracle}(\boldsymbol{\mu})} \ge \lambda \max_i \mu_i$.

Definition 2. *The approximation regret with rate λ of MAB problem on mean vector $\boldsymbol{\mu}$ is defined as:*

$$\sum_{t=1}^T (\lambda \max_i \mu_i - \mu_{i(t)}),$$

where $i(t)$ is the arm pulled on time step t.

The TS algorithm using approximation oracle Oralce works just as Algorithm 1.

Theorem 2. *There exists an MAB instance with an approximation oracle such that when using Algorithm 1, the regret is $\Omega(T)$.*

Proof Sketch. Consider the following MAB instance:

Problem Instance 1. $m = 3$, $\boldsymbol{\mu} = [0.9, 0.82, 0.7]$, *approximate rate* $\lambda = 0.8$. *The* Oracle *works as following: if* $\mu_3 \ge \lambda \max_i \mu_i$, $\mathsf{Oracle}(\boldsymbol{\mu}) = 3$; *else if* $\mu_2 \ge \lambda \max_i \mu_i$, $\mathsf{Oracle}(\boldsymbol{\mu}) = 2$; *otherwise* $\mathsf{Oracle}(\boldsymbol{\mu}) = 1$.

The key idea is that we may never play the best arm (arm 1 above) when using the approximation oracle. When the sample from the prior distribution of the best arm is good, we choose an approximate arm (arm 2 above) but not the best arm; otherwise we choose an bad arm (arm 3 above) with positive approximation regret. Thus the expected regret of each time slot depends on whether the prior distribution of the best arm at the beginning is good or not. Since the best arm is never observed, we never update its prior distribution. Thus the expected regret in each time slot can remain a positive constant forever. □

3.3. The Exponential Constant Term

Since every arm's sample $\theta_i(t)$ is chosen independently, the worst case is that we need all the samples for base arms in the best super arm to be close to their true means to choose that super arm. Under this case, the probability that we have no regret in each time slot is exponentially with k^*, thus we will have such a constant term.

Theorem 3. *There exists a CMAB instance such that the regret of Algorithm 1 on this instance is at least $\Omega(2^{k^*})$.*

Proof Sketch. Consider the following CMAB instance:

Problem Instance 2. $m = k^* + 1$, *there are only two super arms in $\mathcal{I}$, where $S_1 = \{1, 2, \cdots, k^*\}$ and $S_2 = \{k^* + 1\}$. The mean vector $\boldsymbol{\mu}_{S_1} = [1, \cdots, 1]$. The reward function R follows $R(S_1, \boldsymbol{X}) = \prod_{i \in S_1} X_i$, and $R(S_2, \boldsymbol{X}) = 1 - \Delta$, while $\Delta = 0.5$. The distributions D_i are all independent Bernoulli distributions with mean μ_i (since $\mu_i = 1$, the observations are always 1).*

One can show that the expected time until Algorithm 1 plays the optimal super arm S_1 for the first time is $\Omega(2^{k^*})$. □

The exponential term comes from the bad prior distribution at the beginning of the algorithm. In fact, from the proof of Theorem 1 we know that if we can pull each base arm for $\tilde{O}(\frac{1}{\varepsilon^2})$ times at the beginning and then use the CTS policy whose prior distribution at the beginning is the posterior distribution after those observations, then we can reduce the exponential constant term to $O(\frac{m}{\varepsilon^4})$. However, since ε depends on $\Delta_{\min}$, which is unknown to the player, we can not simply run each base arm for a few time steps to avoid the exponential constant regret term. Perhaps an adaptive choice can be used here, and this is a further research item.

4. Matroid Bandit Case

In matroid bandit, we suppose the oracle we use is the greedy one since the greedy algorithm gives back the exact best super arm.

4.1. Regret Upper Bound

Let $S^* \in \operatorname{argmax}_{S\in\mathcal{I}} r(S,\boldsymbol{\mu})$ be one of the optimal super arm. Define $\Delta_i = \min_{j|j\in S^*,\mu_j>\mu_i} \mu_j - \mu_i$. If $i \notin S^*$ but $\{j \mid j \in S^*, \mu_j > \mu_i\} = \emptyset$, we define $\Delta_i = \infty$, so that $\frac{1}{\Delta_i} = 0$. Let $K = \max_{S\in\mathcal{I}} |S| = |S^*|$. We have the following theorem for CTS algorithm under the matroid bandit case:

Theorem 4. *Under Assumptions 1 and 2, the regret upper bound of Algorithm 1 for a matroid bandit is:*

$$Reg(T) \le \sum_{i\notin S^*} \frac{\log T}{\Delta_i - 2\varepsilon}\frac{\Delta_i - \varepsilon}{\Delta_i - 2\varepsilon} + \alpha_2 \cdot \left(\frac{m}{\varepsilon^4}\right) + m^2,$$

for any $\varepsilon > 0$ such that $\forall i \notin S^*, \Delta_i - 2\varepsilon > 0$, where α_2 is a constant not dependent on the problem instance.

Notice that we do not need the distributions of all the base arms to be independent due to the special structure of matroid. When ε is small, the leading $\log T$ term of the above regret bound matches the regret lower bound $\sum_{i\notin S^*}\frac{1}{\Delta_i}\log T$ given in (Kveton et al., 2014). For the constant term, we have an $O(\frac{1}{\varepsilon^4})$ factor while Agrawal & Goyal (2013) have an $O(\frac{1}{\varepsilon^2})$ factor in their theorem. However, even following their analysis, we can only obtain $O(\frac{1}{\varepsilon^4})$ and cannot recover the $O(\frac{1}{\varepsilon^2})$ in their analysis.

We now provide a proof outline. The difference from Theorem 1 is that we can use the special combinatorial structure of matroid to improve the analysis.

Firstly, we introduce a fact from (Kveton et al., 2014).

Fact 1. *(Lemma 1 in (Kveton et al., 2014)) For each $S(t) = \{i^{(1)}(t), \cdots, i^{(K)}(t)\}$ chosen by Algorithm 1 (the superscript is the order when they are chosen), we could find a bijection L_t from $\{1, 2, \ldots, K\}$ to S^* such that:*

1) If $i^{(k)}(t) \in S^*$, then $L_t(k) = i^{(k)}(t)$;

2) $\forall 1 \le k \le K$, $\{i^{(1)}(t), \cdots, i^{(k-1)}(t), L_t(k)\} \in \mathcal{I}$.

With a bijection L_t, we could decouple the regret of playing one action $S(t)$ to each pair of mapped arms between $S(t)$ and S^*. For example, the regret of time t is $\sum_{k=1}^K \mu_{L_t(k)} - \mu_{i^{(k)}(t)}$.

We use $N_{i,j}(t)$ to denote the number of rounds that $i^{(k)}(t) = i$ and $L_t(k) = j$ for $i \notin S^*$, $j \in S^*$ within in time slots $1, 2, \cdots, T$, then

$$Reg(T) \le \sum_{i\notin S^*}\sum_{j:j\in S^*,\mu_j>\mu_i} \mathbb{E}[N_{i,j}(t)](\mu_j - \mu_i).$$

We can see that if $\{j : j \in S^*, \mu_j > \mu_i\} = \emptyset$, then $\sum_{i\notin S^*}\sum_{j:j\in S^*,\mu_j>\mu_i} \mathbb{E}[N_{i,j}(t)](\mu_j - \mu_i) = 0$, thus we do not need to consider the regret from base arm i, so we set $\Delta_i = \infty$ to make $\frac{1}{\Delta_i} = 0$.

Now we just need to bound the value $N_{i,j}(t)$, similarly, we can defined the following three events:

- $\mathcal{A}_{i,j}(t) = \{\exists k, i^{(k)}(t) = i \wedge L_t(k) = j\}$
- $\mathcal{B}_i(t) = \{\hat{\mu}_i(t) > \mu_i + \varepsilon\}$
- $\mathcal{C}_{i,j}(t) = \{\theta_i(t) > \mu_j - \varepsilon\}$

Thus

$$\begin{aligned}
\mathbb{E}[N_{i,j}(t)] &= \mathbb{E}\left[\sum_{t=1}^T \mathbb{I}[\mathcal{A}_{i,j}(t)]\right] \\
&\le \mathbb{E}\left[\sum_{t=1}^T \mathbb{I}[\mathcal{A}_{i,j}(t) \wedge \mathcal{B}_i(t)]\right] \\
&\quad + \mathbb{E}\left[\sum_{t=1}^T \mathbb{I}[\mathcal{A}_{i,j}(t) \wedge \neg\mathcal{B}_i(t) \wedge \mathcal{C}_{i,j}(t)]\right] \\
&\quad + \mathbb{E}\left[\sum_{t=1}^T \mathbb{I}[\mathcal{A}_{i,j}(t) \wedge \neg\mathcal{C}_{i,j}(t)]\right]
\end{aligned}$$

We can use Chernoff Bound to get an upper bound of the first term as $(m-K)(1+\frac{1}{\varepsilon^2})$. As for the second term, basic properties of Beta distribution give an upper bound: $(m-K)K + \sum_{i\notin S^*}\frac{\log T}{2\Delta_i^2}$.

The largest difference appears in the third term, instead of Lemma 1, here we can have some further steps in matroid bandit.

Lemma 2. *Suppose the vector $\boldsymbol{\theta}(t)$ satisfy that $\mathcal{A}_{i,j}(t) \wedge \neg\mathcal{C}_{i,j}(t)$ happens. Then if we change $\theta_j(t)$ to $\theta'_j(t) > \mu_j - \varepsilon$ and set other values in $\boldsymbol{\theta}(t)$ unchanged to get $\boldsymbol{\theta}'(t)$, then arm j must be chosen in $\boldsymbol{\theta}'(t)$.*

We use the notation $\boldsymbol{\theta}_{-i}$ to be the vector $\boldsymbol{\theta}$ without θ_i.

For any $j \in S^*$, let W_j be the set of all possible values of $\boldsymbol{\theta}$ satisfies that $\mathcal{A}_{i,j}(t) \wedge \neg\mathcal{C}_{i,j}(t)$ happens for some i, and $W_{-j} = \{\boldsymbol{\theta}_{-j} : \boldsymbol{\theta} \in W_j\}$.

From Lemma 2, we know $\boldsymbol{\theta}(t) \in W_j$ only if $\boldsymbol{\theta}_{-j}(t) \in W_{-j}$ and $\theta_j(t) \le \mu_j - \varepsilon$. Then similar with the analysis of Theorem 1, we can bound the value $\sum_{t=1}^{T} \mathbb{E}[\boldsymbol{\theta}(t) \in W_j] \le O(\frac{1}{\varepsilon^4})$.

5. Experiments

We conduct some preliminary experiments to empirically evaluate the performance of CTS versus CUCB and C-KL-UCB. The reason that we choose C-KL-UCB is that: a) in classical MAB model, KL-UCB behaves better than UCB; b) similar with TS, it is also a policy based on Bayesian Rule. We also make simulations on CUCB and C-KL-UCB with chosen parameters, represented by CUCB-m and C-KL-UCB-m. In CUCB, we choose the confidence radius to be $\mathrm{rad}_i(t) = \sqrt{\frac{3\log t}{2N_i(t)}}$, while in CUCB-m, it is $\sqrt{\frac{\log t}{2N_i(t)}}$. In C-KL-UCB we choose $f(t) = \log t + 2\log\log t$, while in C-KL-UCB-m it is $\log t$. Those chosen parameters in CUCB-m and C-KL-UCB-m make them behave better, but lack performance analysis.

5.1. Matroid Bandit

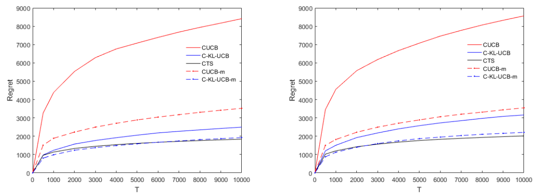
It is well known that spanning trees form a matroid. Thus, we test the maximum spanning tree problem as an example of matroid bandits, where edges are arms, and super arms are forests.

We first generate a random graph with M nodes, and each pair of nodes has an edge with probability p. If the resulting graph has no spanning tree, we regenerate the graph again. The mean of the distribution is randomly and uniformly chosen from $[0, 1]$. The expected reward for any spanning tree is the sum of the means of all edges in it. It is easy to see that this setting is an instance of the matroid bandit.

The results are shown in Figure 1 with the probability $p = 0.6$ and $M = 30$. In Figure 1(a), we set all the arms to have independent distributions. In Figure 1(b), each time slot we generate a global random variable $rand$ uniformly in $[0, 1]$, all edges with mean larger than $rand$ will have outcome 1, while others have outcome 0. In other words, the distributions of base arms are correlated. We can see that CTS has smaller regret than CUCB, CUCB-m and C-KL-UCB in both two experiments. As for C-KL-UCB-m algorithm, it behaves better with small T, but loses when T is very large. We emphasize that C-KL-UCB-m policy uses parameters without theoretical guarantee, thus CTS algorithm is a better choice.

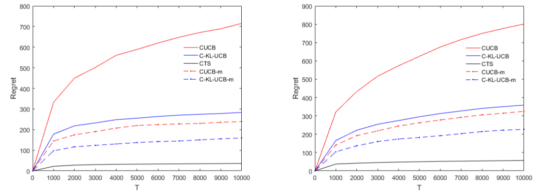
5.2. General CMAB

In the general CMAB case, we consider the shortest path problem. We build two graphs for this experiment, the results of them are shown in Figure 2(a) and Figure 2(b). The cost of a path is the sum of all edges' mean in that path, while the outcome of each edge e follows a independent Bernoulli distribution with mean μ_e. The objective is to find the path with minimum cost. To make the problem more challenging, in both graphs we construct a lot of paths from the source node s to the sink node t that only have a little larger cost than the optimal one, and some of them are totally disjoint with the optimal path.



(a) (b)

Figure 1. Experiments on matroid bandit: Maximum Spanning Tree



(a) (b)

Figure 2. Experiments on general CMAB: Shortest Path

Similar to the case of matroid bandit, the regret of CTS is also much smaller than that of CUCB, CUCB-m and C-KL-UCB, especially when T is large. As for the C-KL-UCB-m algorithm, although it behaves best in the four UCB-based policies, it still has a large difference between CTS.

6. Future Work

In this paper, we apply combinatorial Thompson sampling to combinatorial multi-armed bandit and matroid bandit problems, and obtain theoretical regret upper bounds for those two settings.

There are still a number of interesting questions that may worth further investigation. For example, pulling each base arm for a number of time slots at the beginning of the game can decease the constant term to non-exponential, but the point is that the player does not know how many time slots are enough. Thus how can we use an adaptive policy or some further assumptions to do so is a good question. In this paper, we suppose that all the distributions for base arms are independent. Another question is how to find analysis for using CTS under correlated arm distributions.

References

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, pp. 39–1, 2012.

Agrawal, S. and Goyal, N. Further optimal regret bounds for thompson sampling. In *AISTATS*, pp. 99–107, 2013.

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 non-stochastic multi-armed bandit problem. *Siam Journal on Computing*, 32(1):48–77, 2002b.

Berry, D. A. and Fristedt, B. *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*. Springer, 1985.

Chen, W., Hu, W., Li, F., Li, J., Liu, Y., and Lu, P. Combinatorial multi-armed bandit with general reward functions. In *NIPS*, 2016a.

Chen, W., Wang, Y., Yuan, Y., and Wang, Q. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016b. A preliminary version appeared as Chen, Wang, and Yuan, “combinatorial multi-armed bandit: General framework, results and applications”, ICML’2013.

Combes, R., Talebi, M. S., Proutiere, A., and Lelarge, M. Combinatorial bandits revisited. In *NIPS*, 2015.

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. *Mathematics*, 2011.

Gittins, J. Multi-armed bandit allocation indices. wiley-interscience series in systems and optimization. 1989.

Gopalan, A., Mannor, S., and Mansour, Y. Thompson sampling for complex online problems. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014a.

Gopalan, A., Mannor, S., and Mansour, Y. Thompson sampling for complex online problems. In *ICML*, volume 14, pp. 100–108, 2014b.

Kaufmann, E., Korda, N., and Munos, R. Thompson sampling: an asymptotically optimal finite-time analysis. In *Proceedings of the 23rd international conference on Algorithmic Learning Theory*, pp. 199–213, 2012.

Komiyama, J., Honda, J., and Nakagawa, H. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1152–1161, 2015.

Kveton, B., Wen, Z., Ashkan, A., Eydgahi, H., and Eriksson, B. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

Kveton, B., Wen, Z., Ashkan, A., and Szepesvári, C. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015a.

Kveton, B., Wen, Z., Ashkan, A., and Szepesvari, C. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems*, pp. 1450–1458, 2015b.

Lai, T. L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1): 4–22, 1985.

Russo, D. and Van Roy, B. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Wang, Q. and Chen, W. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Advances in Neural Information Processing Systems*, 2017.

Wen, Z., Kveton, B., and Ashkan, A. Efficient learning in large-scale combinatorial semi-bandits. In *ICML*, pp. 1113–1122, 2015.