Regret Analysis of Stochastic Multi-armed Bandit Problem with Clustered Information Feedback

Tianchi Zhao*
Department of ECE
University of Arizona,
Tucson, AZ
tzhao7@email.arizona.edu

Bo Jiang*
Department of ECE
University of Arizona,
Tucson, AZ
bjiang@email.arizona.edu

Ming Li
Department of ECE
University of Arizona,
Tucson, AZ
lim@email.arizona.edu

Ravi Tandon
Department of ECE
University of Arizona,
Tucson, AZ
tandonr@email.arizona.edu

***Abstract*—In this paper, we analyze the regret bound of Multi-armed Bandit (MAB) algorithms under the setting where the payoffs of an arbitrary-size cluster of arms are observable in each round. Compared to the well-studied bandit or full feedback setting, where the payoffs of the selected arm or all the arms are observable, the clustered feedback setting can be viewed as a generalization and a connection. We focus on two most representative MAB algorithms: Upper Confidence Bound and Thompson sampling, and adapt them into the clustered feedback setting. Then, we theoretically derive the regret bound for each of them considering the general type of payoffs (value comes from continuous domains). We show that the regret bounds of these two algorithms with clustered information feedback depend only on the number of clusters. Finally, we simulate both synthetic data and real-world data to compare the performance of these algorithms with different numbers of observable payoffs in each round, the results validate our analysis.**

***Index Terms*—multi-armed bandit, regret bound, clustered information feedback, problem-dependent bound.**

I. Introduction

A multi-armed bandit problem is a sequential allocation problem defined by a set of actions, which can be viewed as a dynamic program without dynamic state information other than the belief state (a state-less version of reinforcement learning (RL) [17]). Compared to other RL algorithms such as Q-learning, The algorithms of MAB are based on more rigors theoretical foundation and provable worst-case performance (regret bounds). The multi-armed bandit (MAB) model focuses on the essential issue of addressing the trade-off between exploration and exploitation, and this is the balance between staying with the option that gave highest payoffs in the past and exploring new options that might give higher payoffs in the future. In the settings of a stochastic MAB model, there are several selective actions (arms) available, and each action's reward follows a particular distribution with an unknown mean. At each time step, a unit resource is allocated to an action and some observable payoff is obtained based on the feedback mechanism.

Depending on different feedback settings, the observation at each round varies, and there are basically two different feedback settings in the research community: Bandit (information) feedback and full (information) feedback. Bandit feedback means only the reward of the selected action is observed at each round [15]. For example, a gambler at a row of slot machines, who observes the reward of the pulled arm at each round. Full feedback [15] means the decision-maker observes the rewards from all actions by selecting one action at each time step, thus, the observations do not depend on the selected action. One real-life scenario with full feedback is investments on a stock market: Suppose each morning, the investor chooses one stock and invests $1 into it. At the end of the day, we observe not only the price of the chosen stock but the prices of all stocks. Based on this feedback, we determine which stock to invest for the next day. Different from these two extreme settings, in this paper, we consider another cluster (information) feedback setting, where the rewards of a bounded set of actions are returned at each round. A real-world example is the online shopping recommendation system: the user could buy an item and rate it. Meanwhile, he would be interested in other items which are similar to the purchased one. These clicks can be regarded as virtual rating. In this scenario, the purchased item and clicked items can be viewed as in the same cluster.

To tackle the MAB problems, many algorithms have been proposed to maximize the total payoffs obtained in a sequence of allocations [6]. In this paper, we study two most representative algorithms: Upper Confidence Bound (UCB_1) algorithm [3] and Thompson Sampling (TS) [16]. The UCB_1 algorithm is a frequentist approach which considers capturing the knowledge about the reward generating process by a set of random variables, and the estimated means (or a similar quantity) of the random variables reflect the current knowledge of the algorithm in a condensed form and guide further exploitation. The widths of the confidence bounds reflect the uncertainty of the algorithm's knowledge and will guide further exploration. On the other hand, TS is a Bayesian approach which consists in choosing the action that maximizes the expected reward with respect to a randomly drawn belief, and updates the posterior using the observed payoff. In other words, TS chooses the action to maximize the expected reward given the sampled parameters, the action and current context.

To evaluate the performance of bandit algorithms, expected

* Equal contribution. This work was partly supported by ONR grant N00014-16-1-2650, NSF grants CNS-1564477, CAREER grant 1651492, CNS-1715947 and the Keysight Early Career professor Award.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

cumulative regret [2] is usually applied which depends on the distance between the mean reward of the optimal action and the chosen action. There are basically two ways to measure the regret in each round: The first way captures the regret by assuming binary reward, where the returned reward in each round only tells if the selected action has succeeded or not. The advantage of this binary reward is a significant reduction of computation complexity. However, this approach fails to capture the distance of the selected action from the optimal one which is useful for guidance of further strategies. The other approach considers a more general setting, where the reward value comes from a continuous domain, which means the regret depends on the distance between the optimal action and the selected action. To bound the cumulative regrets, bounds considering binary reward value is usually denoted as problem-independent bounds; bounds considering general reward value are denoted as problem-dependent bounds [2]. In this paper, we study problem-dependent bounds.

The main contributions of this paper are three-fold:

1) We adapt the UCB_1 and TS algorithms with general reward values under the clustered feedback setting, where a cluster of actions' rewards are observed in each round.
2) We theoretically derive the problem-dependent cumulative regret bounds for the proposed UCB_1 and TS algorithms under the clustered feedback setting. We show that the proposed algorithms and regret bounds can be reduced to bandit or full feedback as two special cases.
3) We simulate with both synthetic data and real-world data to compare the performance of the two algorithms with different cluster sizes, and the results testify our bounds.

The remainder of the paper is organized as follows. In Section II, we introduce related works. In Section III, we present our model with clustered feedback setting, and show how to adapt the UCB_1 and TS algorithms into the proposed model. In Section IV, we derive cumulative regret bounds for UCB_1 and TS algorithms. In Section V, we show our simulation results, and summarize key insights. In Section VI, we offer concluding remarks.

II. Background and Related Work

The expected cumulative regret can be expressed as:

$$E[R(T)] = \sum_{t=1}^{T} (\mu_{I^*} - \mu_{I_t}), \tag{1}$$

where μ_{I^*} denotes the mean reward of the optimal action, μ_{I_t} denotes the mean reward of the action chosen at t, T denotes the time horizon.

Next, we present the original UCB_1 and TS algorithms under the bandit feedback setting followed by the their cumulative regrets.

A. UCB_1 algorithm

The basic idea of the UCB_1 algorithm can be described as follows: Firstly, assign each action with an upper Confidence bound (UCB), and then, select the action with the highest UCB. Finally, update the UCB of each action according to the observed reward, the number of times this action has been selected as well as the total times. Specially, Denote $r_t(i)$ as the confidence radius for an action i at time t. Let $n_t(i)$ be the number of times action i has been selected up to rounds t, and $\hat{\mu}_t(i)$ is the average reward of action i up to time t. The upper confidence bound of arm i at time t is defined as:

$$\text{UCB}_{1t}(i) = \hat{\mu}_t(i) + r_t(i), \tag{2}$$

where $r_t(i) = \sqrt{\frac{2ln(T)}{n_t(i)}}$ is applied to encourage exploration. Then, the UCB_1 algorithm chooses the best action based on the optimistic estimate. The algorithm is described in algorithm 1:

Algorithm 1 UCB_1 algorithm [4]

$\hat{\mu}_t(i) = 0, n_t(i) = 0$
Select each arm at least once and update $\hat{\mu}_t(i), n_t(i)$ accordingly
//Main Loop
while 1 do
Select arm i that maximizes $\hat{\mu}_t(i) + r_t(i)$
Update $\hat{\mu}_t(i), n_t(i)$ for arm i
end while

Auer *et al.* show that under the UCB_1 algorithm, after T actions, the expected regret is upper bounded by [4]:

$$E[R_T] \leq \sum_{i=2}^{N} \frac{8lnT}{\Delta_i} + 4 \sum_{i=2}^{N} \Delta_i, \tag{3}$$

which depends on the number of N, the maximal distance between the reward of the optimal arm and sub-optimal arm, as well as the time horizon T.

B. Thompson sampling algorithm

Thompson sampling is an online learning algorithm that has been widely applied to many decision-making problems [5], [10], [17]. The basic idea of Thompson sampling can be summarized as follows: It is assumed that the success probability or mean reward of each action follows a certain distribution (with different means). In each round, the decision-maker firstly samples each action's reward according to the prior distribution and then selects the one with the highest sampled reward, he then updates corresponding posterior with the reward $r_t \in [0, 1]$. Usually, the Beta distribution is adopted as the prior of the success probability in the Bernoulli trial since it is the conjugate distribution. The process of the Thompson sampling algorithm is described in algorithm 2.

Agrawal [2] gives a problem-dependent regret bound for the Thompson sampling algorithm with a statement of: for the N-armed stochastic bandit problem, for any constant $0 < \epsilon \leq 1$, Thompson sampling algorithm has an expected regret of

$$E[R_T] \leq (1+\epsilon)^2 \sum_{i=2}^{N} \frac{lnT}{d(\mu_i, \mu_1)} \Delta_i + O\left(\frac{N}{\epsilon^2}\right), \tag{4}$$

Algorithm 2 Thompson sampling algorithm [1]

$S_i = 0, F_i = 0$
for $t = 1, 2, ...,$ **do**
  For each arm $i = 1, \ldots, N$, sample $\theta_i(t)$ from the Beta($S_i + 1$,$F_i + 1$) distribution
  Play arm $i(t) := \arg\max \theta_i(t)$ and observe reward r_t
  if $r_t = 1$ **then**
    $S_i = S_i + 1$
  else
    $F_i = F_i + 1$
  end if
end for

where T is the number of rounds played, μ_1 is the mean of the optimal action (without loss of generality, the first possible action is assumed to be the optimal), and $d(\mu_i, \mu_1)$ is the KL-divergence between Bernoulli distributions of μ_i and μ_1. $\Delta_i = \mu_1 - \mu_i$. According to Eq. (4), we can see that, under the bandit feedback setting and for the general reward values, the expected regret bound mainly depends on the number of arms and the distance of mean values between the optimal action and sub-optimal actions.

From Equation (2) (4), we can see that the expressions of both regrets include a summation of the number of actions. The reason is that, the algorithm only observes the reward of the action selected at time t. So, the cumulative regret includes the regret of all sub-optimal actions.

Russo and Van Roy show that the problem-independent cumulative regret bound for Thomson sampling under full information setting does not depend on the number of actions [15], which can be expressed as:

$$E[R(T)] \leq \sqrt{\frac{1}{2} H(A^*) T}, \tag{5}$$

where $H(A^*)$ is the entropy of the optimal action A^*. Compared to the result in the bandit information feedback,

$$E[R(T)] \leq \sqrt{\frac{1}{2} |N| H(A^*) T}.$$

However, in [15], only a problem-independent case with binary reward domain was studied, also, they did not give regret bound for other bandit algorithms. However, such observation that whether the regret bound include the summation of arms depends on the feedback setting motivates our work to study in depth of the problem-dependent bounds as well as deriving the bounds under other feedback settings.

C. Extensions of UCB_1 and TS

There are also many other algorithms trying to solve the MAB problems which can be viewed as extensions of UCB and TS. Olivier *et al.* [7] present a KL-UCB algorithm, which differs from UCB in the measurement of the confidence of the empirical mean. Meanwhile, experimental results [8] and theoretical analysis [12] show that it reaches the lower bound of Lai and Robbins [14]. Kaufmann *et al.* [11] present Bayesian upper confidence bounds algorithm (Bayes-UCB), where the arm selection is determined by the quantiles of the posterior distribution. They also give a finite-time regret bound in the order of $O(N \ln(T))$ for the binary reward case.

The aforementioned MAB algorithms assume that only one arm can be played at each time step. The following bandit problems assume the agent may play L arms simultaneously. The first such model is called Multi-armed bandits with Multiple Plays (MAB-MP) [18]. For a N armed MAB-MP problem, the agent selects the top L arms to obtain the largest cumulative reward value. Another multi-play problem is called combinatorial Multi-Armed bandits (CMAB) [13]. In a N armed CMAB problem, and the agent needs to pull a set of base arms S in each round, where the combination of S is called a super arm. The feedback is the summation of all the rewards of the base arms, and can be viewed as the reward of the super arm. For example, in a routing problem, the agent chooses a super arm (routing strategy). Each super arm contains L base arms (links). The difference between multi-play and cluster feedback lies in whether the arms whose rewards are observable are predefined. In multi-play problems, the agent chooses multiple arms and each returned reward contribute to the cumulative regret. In cluster feedback problems, the returned rewards are predefined by clusters, the agent selects one arm at each time and the regret depends only on the reward of this arm, though multiple rewards of arms within the same cluster are observable. Zhao *et al.* [19] propose a hierarchical Thompson sampling (HTS), which divides arms into clusters. A cluster is firstly sampled and one arm inside the cluster is specified. However, HTS only observes the selected arm's reward and hence, is different from clustered feedback setting. To the best of our knowledge, we are the first to tackle clustered feedback settings.

III. Problem Statement and Generalized UCB_1 and TS algorithms

In this section, we state our model setup and clarify all the notations used throughout this paper, and then we present the adapted versions of UCB_1 and TS algorithms.

TABLE I
List of symbols

N	Total number of arms	i	index of arm
μ_i	mean reward of arm i	t	index of time step
$n_t(C_j)$	Times of choosing C_j till t	T	Time Horizon
$\theta_i(t)$	sampled reward of arm i at t	C_j	cluster j
N_C	Total number of clusters	L_j	size of cluster j
$R(T)$	Cumulative regret till T	$r(i)$	reward of arm i
$\hat{\mu}_t(i)$	empirical mean reward of i at t	Δ_i	$\mu_1 - \mu_i$

A. Problem Statement

It is assumed that there are N arms (actions) available in the model, the reward of each arm follows a certain probability distribution which is unknown to the agent. Denote μ_i as the mean reward of the i-th arm. At each time step, the agent pulls an arm whose reward is returned immediately, also, the rewards of several related arms are observed simultaneously.

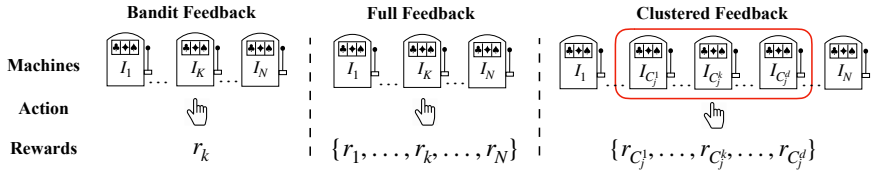

Fig. 1. Feedback settings considered in this paper.

The returned set of arms is predefined by clusters which we will define later. Denote $i(t)$ as the index of the selected arm at time t, and $r(i_t) \in [0,1]$ as the observed reward of the i-th arm at time t, $1 \leq t \leq T$, where T is the time horizon. The agent wants to select the optimal arm (the one with highest mean value) in order to maximize the cumulative payoff or minimize the cumulative regret, without loss of generality, in the following, denote the first arm $i = 1$ as the optimal arm, and denote Δ_i as the distance between the mean reward of arm 1 and arm i: $\Delta_i = \mu_1 - \mu_i$.

Next, we define the clustered feedback setting, which is captured by the notion of clusters. Denote C_j as a cluster which contains L_j arms ($1 \leq L_j \leq N$), N_c as the total number of clusters. As a result, $\sum_{j=1}^{N_c} L_j = N$. Under the clustered information feedback setting, it is assumed that, if the i-th arm belongs to the j-th cluster, after pulling the i-th arm, all the rewards of arms in the j-th cluster are returned. Preliminary works about cluster applied in MAB problems can be found in [2] and [4], however, our model assumes L_j may be different from cluster to cluster thus can be viewed as the most general one. Also by adjusting the size of each cluster, we are able to reduce the cluster setting to the other two settings: when each cluster only contains one possible arm, the feedback setting becomes the bandit feedback; on the other hand, if there is only one cluster holding all the arms, it becomes full feedback setting. The feedback settings are illustrated in Fig. 1.

In the following sections, without loss of generality, it is assumed that C_1 contains the global optimal arm 1. Notations used throughout this paper are summarized in Table I.

B. UCB_1 *and TS under Clustered Information Feedback*

The original UCB_1 and TS algorithms are studied either in bandit or full feedback settings. Next, we adapt the two algorithms into the clustered information feedback setting with general reward values.

The algorithm of the UCB_1 algorithm under clustered information feedback is shown in algorithm 3. Compared to the original UCB_1 algorithm, the only difference in Algorithm 3 is the number of updated distributions due to the feedback setting. Note that, $n_t(C_j)$ denotes the times that the algorithm chooses an arm $i \in C_j$ before t, which also means the number of times that cluster C_j has been chosen. Under the cluster setting, when $i \in C_j$ is selected, all the rewards of arm $i' \in C_j$ are returned, which ensures the empirical mean of $\hat{\mu}_{t-1}(i')$ to be an unbiased estimation of $\mu(i')$. Note that, the "exploration term" becomes $r_t(i) = \sqrt{\frac{2ln(t)}{n_t(C_j)}}$, where the denominator is determined by $n_t(C_j)$ instead of $n_t(i)$. This term is derived from Chernoff-Hoeffding inequality, which provides an upper bound on the probability of the sum of a random variable deviating from its expected value. Note that the arm $i \in C_j$ updates its reward when any arm $i' \in C_j$ is selected, as a result, $r_t(i)$ depends on the total number of times that the cluster j has been selected.

Algorithm 3 UCB_1 algorithm with clustered feedback

$\hat{\mu}_t(i) = 0$, $n_t(C_j) = 0$, $\forall C_j$
Select each arm at least once and update $\hat{\mu}_t(i), n_t(C_j)$ accordingly
//Main Loop
while 1 do
Select arm i that maximizes $\hat{\mu}_t(i) + r_t(i)$
for $\forall C_j$ **do**
  if $i \in C_j$ **then**
    for $\forall i' \in C_j$ **do**
      Observe the reward $r(i') \in [0,1]$
      Update $\hat{\mu}_t(i') = \frac{\hat{\mu}_{t-1}(i')n_{t-1}(C_j)+r(i')}{n_{t-1}(C_j)+1}$
      Update $n_t(C_j) = n_{t-1}(C_j) + 1$
      Update $r_t(i') = \sqrt{\frac{2ln(t)}{n_t(C_j)}}$
    end for
  end if
end for
end while

The stochastic TS algorithm under clustered information feedback is given in Algorithm 4. Since the probability of observing success in the Bernoulli trial is equal to its mean reward μ_i. Let f_i denote the pdf of the reward distribution for the arm i. Then, the general reward with value from continous domain can be converted to the Bernoulli distribution by [1]

$$Pr(r_t(i) = 1) = \int_0^1 \tilde{r}_t f_i(\tilde{r}_t) d\tilde{r}_r(i) = \mu_i. \tag{6}$$

Note that, in Algorithm 4, there is an additional sampling which transfers the continuous reward into a binary reward, so that, the posterior belief of each arm can be updated by Beta distribution. Besides that, the posteriors of all the arms within cluster C_j are updated.

IV. Regret Bounds for UCB_1 and TS under Clustered Feedback Setting

In this Section, we theoretically derive the regret bounds for UCB_1 and TS algorithms under the clustered information

Algorithm 4 Thompson sampling with clustered feedback

$S_i = 0, F_i = 0$
for $t = 1, 2, ...,$ **do**
  For each arm $i = 1, \ldots, N$, sample $\theta_i(t)$ from the Beta($S_i + 1$,$F_i + 1$) distribution
  Play arm $i(t) := \arg\max \theta_i(t)$
  for $\forall C_j$ **do**
    if $i(t) \in C_j$ **then**
      for $\forall i' \in C_j$ **do**
        Observe reward $\tilde{r}(i') \in [0, 1]$
        Perform a Bernoulli trial with success probability $\tilde{r}(i')$ and observe output $r(i') \in \{0, 1\}$
        if $r(i') = 1$ **then**
          $S_{i'} = S_{i'} + 1$
        else
          $F_{i'} = F_{i'} + 1$
        end if
      end for
    end if
  end for
end for

feedback setting, and show that they achieve logarithmic regret (depending on the number of clusters).

For the i-th arm, define two thresholds between μ_i and μ_1.

Definition 1 (Thresholds x_i, y_i)**.** *we choose* $\mu_i < x_i < y_i < \mu_1$ *as follows:* $\mu_i < x_i < \mu_1$ *such that* $d(x_i, \mu_1) = \frac{d(\mu_i,\mu_1)}{1+\epsilon}$. $x_i < y_i < \mu_1$ *such that* $d(x_i, y_i) = \frac{d(x_i,\mu_1)}{1+\epsilon} = \frac{d(\mu_i,\mu_1)}{(1+\epsilon)^2}$, $0 < \epsilon < 1$.

Then, the problem-dependent expected cumulative regret which both algorithms minimize can be expressed as:

$$
\begin{aligned}
&E[R(T)] \\
=&E\left[\sum_{i\notin C_1}\sum_{t=1}^{T}\Delta_i \mathbb{1}_{I(t)=i}\right] + E\left[\sum_{i\in C_1, i\neq 1}\sum_{t=1}^{T}\Delta_i \mathbb{1}_{I(t)=i}\right] \\
=&E\left[\sum_{C_j\neq C_1}\sum_{i\in C_j}\sum_{t=1}^{T}\Delta_i \mathbb{1}_{I(t)=i}\right] + E\left[\sum_{i\in C_1, i\neq 1}\sum_{t=1}^{T}\Delta_i \mathbb{1}_{I(t)=i}\right] \\
\leq&E\left[\sum_{C_j\neq C_1}\sum_{i\in C_j}\sum_{t=1}^{T}\Delta_{C_j} \mathbb{1}_{I(t)=i}\right] + E\left[\sum_{i\in C_1, i\neq 1}\sum_{t=1}^{T}\Delta_{C_1} \mathbb{1}_{I(t)=i}\right] \\
=&\sum_{C_j\neq C_1}\Delta_{C_j} E\left[\sum_{i\in C_j}\sum_{t=1}^{T}\mathbb{1}_{I(t)=i}\right] + \Delta_{C_1} E\left[\sum_{i\in C_1, i\neq 1}\sum_{t=1}^{T}\mathbb{1}_{I(t)=i}\right].
\end{aligned} \tag{7}
$$

Where $\Delta_{C_j} = \max_{i\in C_j}\{\mu_1 - \mu_i\}$. In the above, we divide the event into two cases with respect to different scenarios of selecting sub-optimal arm: **Case 1**, selecting the sub-optimal arm $i \in C_j, C_j \neq C_1$, and **Case 2**, selecting the sub-optimal arm $i \in C_1$, where $i \neq 1$. Next, we derive regret bounds for UCB_1 and TS according to these two cases.

The basic ideas of the following proofs are as follows: Under each algorithm, we specify the events that cause the selection of a sub-optimal arm for each of the two cases described above. Then, we upper bound the probability of the occurrence of each event. Finally, by combining all the upper bounds of events, we derive the upper bound of the cumulative regrets of each algorithm.

A. UCB_1 *algorithm*

For the UCB_1 algorithm, the sub-optimal arm $i \neq 1$ will be pulled in two cases: either arm 1 and arm i have been insufficiently sampled so that their empirical means are indistinguishable, or the upper confidence bounds derived from Chernoff-Hoeffding's inequality fails. We begin by bounding the probability that a sub-optimal arm is selected due to insufficient sampling. Suppose that there are two events: $A_t(i) : \hat{\mu}_i(t) \leq \mu_i + r_t(i)$. $B_t(i) : \hat{\mu}_1(t) \geq \mu_1 - r_t(1)$. Applying Chernoff-Hoeffding's inequality to bound the probabilities of the complements of events $A_t(i)$ and $B_t(i)$. For the $A_t^c(i)$, we have:

$$
Pr(A_t^c(i)) = Pr(\hat{\mu}_i(t) - \mu_i > \epsilon) \leq e^{\left(\frac{-2\epsilon^2 t^2}{\sum_{i=1}^{t}(1-0)^2}\right)} = e^{-2\epsilon^2 t}. \tag{8}
$$

Plug in the bounding value $\epsilon = r_t(i)$, then $Pr(A_t^c(i)) \leq t^{-2}$.

Similarly, $Pr(B_t^c(i)) \leq t^{-2}$. According to [4], the sub-optimal arm i is pulled at most $\frac{8ln(T)}{\Delta_i^2}$ times when A_t and B_t hold which means i is selected if either it has not been sampled sufficiently (less than $\frac{8ln(T)}{\Delta_i^2}$) or either event A_t or B_t fails. According to the algorithm 3, when an arm $i \in C_j$ is pulled, the rewards of all arms in the cluster C_j are updated simultaneously. We choose $M_{C_j}(T) = \max_{i\in C_j}\{\frac{8ln(T)}{\Delta_i^2}\}$ to ensure all arms $i \in C_j$ have been pulled for sufficient times, which means the arm with largest reward in C_j could be distinguished from the optimal arm 1 then all the other sub-optimal arms in C_j could be distinguishable.

We next bound the two cases defined in Eq. (7).

1) ***Case 1****:* $i \in C_j, C_j \neq C_1$*:* The next lemma states the upper bound of the expected number of selected sub-optimal arms in **Case 1**.

Lemma 1. *The expected number of pulled sub-optimal arms from Cluster* C_j *is upper bounded by* $M_{C_j}(T) + 4|C_j|$.

Proof.

$$
\begin{aligned}
&E\left[\sum_{t=1}^{T}\sum_{i\in C_j}\mathbb{1}_{I(t)=i}\right] \\
\leq& M_{C_j}(T) + \sum_{i\in C_j}\sum_{t=1}^{T} Pr(A_t^c(i) \cup B_t^c(i)) \\
\leq& M_{C_j}(T) + \sum_{i\in C_j}\sum_{t=1}^{T} [Pr(A_t^c(i)) + Pr(B_t^c(i))] \\
\overset{(a)}{\leq}& M_{C_j}(T) + \sum_{i\in C_j}\sum_{t=1}^{T}(t^{-2} + t^{-2}) \leq M_{C_j}(T) + 4|C_j|.
\end{aligned} \tag{9}
$$

Inequality (a) is based on the fact that $\sum_{t=1}^{T} t^{-2} \leq 1 + \int_1^\infty x^{-2}dx = 1 + \frac{-1}{1-2} = 2$. □

*2) **Case 2**: $i \in C_1, i \neq 1$:* The upper bound of the expected sub-optimal selections under **Case 2** follows the next lemma.

Lemma 2. *The expected number of pulled sub-optimal arms from Cluster C_1 is upper bounded by $M_{C_1}(T) + 4|C_1 - 1|$.*

The proof is similar to that of **Case 1**. The only difference is the number of sub-optimal arms. Combining the upper bounds of **Case 1** and **Case 2**, we get the upper bound for UCB_1 under clustered feedback setting, which follows the next Theorem.

Theorem 1. *For the N_C-cluster stochastic bandit problem, if we follow the UCB_1 procedure given in Algorithm 3, the expected regret is:*

$$E[R(T)] \leq \sum_{j=1}^{N_C} \Delta_{C_j} M_{C_j}(T) + 4 \sum_{j=2}^{N_C} |C_j| \Delta_{C_j} + 4\Delta_{C_1}|C_1 - 1|. \tag{10}$$

Remark 1. *This expected cumulative regret for UCB_1 with clustered feedback depends on the number of clusters N_C. Under the full information feedback setting, $N_C = 1$, and the regret becomes:*

$$E[R(T)] \leq \Delta_{\max} M(T) + 4(N-1)\Delta_{\max}. \tag{11}$$

where $M(T) = \max_{i \in \{1,2,\ldots,N\}} \left\{ \frac{8ln(T)}{(\mu_1 - \mu_i)^2} \right\}$, $\Delta_{\max} = \max_{i \in \{1,2,\ldots,N\}} \{\mu_1 - \mu_i\}$.

On the other hand, under the bandit information feedback setting, the $N_C = N$, take in the values of $M(T)$, we have:

$$E[R_T] \leq \sum_{i=2}^{N} \frac{8lnT}{\Delta_i} + 4 \sum_{i=2}^{N} \Delta_i. \tag{12}$$

Which is identical to eq.(3).

B. Thompson sampling for clustered feedback:

We next analysis the Thompson sampling algorithm under the clustered feedback setting. First, we analysis the case $i \in C_j, C_j \neq C_1$ and divide the event into three sub-events which cause the selection of a sub-optimal arm. Then, we analysis the case $i \in C_1, i \neq 1$ and divide the event into sub-events similar to the Case 1.

*1) **Case 1**: $i \in C_j, C_j \neq C_1$:* For TS, we divide the event **Case 1** into three sub-events which are denoted as E_1, E_2 and E_3, and each of them can be expressed as:

$$\begin{aligned} E_1 =& Pr(I(t) = i, \hat{\mu}(t) > x_i), \\ E_2 =& Pr(I(t) = i, \hat{\mu}(t) \leq x_i, \theta_i(t) \geq y_i), \\ E_3 =& Pr(I(t) = i, \hat{\mu}(t) \leq x_i, \theta_i(t) \leq y_i). \end{aligned} \tag{13}$$

Where **Event 1** considers the cases where the empirical mean $\mu_i(t)$ is much greater than its expectation. **Event 2** considers the cases where the sampled value $\theta_i(t)$ is much greater than its expectation. **Event 3** considers the cases where both the empirical mean $\mu_i(t)$ and the sampled value $\theta_i(t)$ are not much greater than its expectations. Then, the regret in one cluster C_j can be expressed as

$$E\left[\sum_{i \in C_j} \sum_{t=1}^{T} \mathbb{1}_{I(t)=i} \right] = \sum_{i \in C_j} \sum_{t=1}^{T} \{E_1 + E_2 + E_3\}. \tag{14}$$

The upper bound of the cumulative regret for **Case 1** follows the next lemma:

Lemma 3. *The cumulative regret for the cluster $C_j \neq C_1$ is upper bounded by*

$$\Delta_{C_j} M'_{C_j}(T) + O(\frac{|C_j|}{\epsilon^2}), \tag{15}$$

where $M'_{C_j}(T) = \max_{i \in C_j} \{(1+\epsilon)^2 \frac{ln(T)}{d(\mu_i, \mu_1)}\}$.

Proof. **Event 1:** Define $\tau_{i,k}$ as the k-th time when $I(t) = i$.

$$\begin{aligned} &\sum_{t=1}^{T} Pr(I(t) = i, \hat{\mu}(t) > x_i) \leq 1 + \sum_{k=1}^{T-1} Pr[\hat{\mu}(\tau_{i,k}) > x_i] \\ &\leq 1 + \sum_{k=1}^{T-1} e^{-kd(x_i, \mu_i)} \leq \frac{1}{d(x_i, \mu_i)} + 1. \end{aligned} \tag{16}$$

This proof follows Agrawal's [2] Lemma 2.

Event 2: Denote k_{C_j} as the number of times that the algorithm chooses an arm $i \in C_j$. Note that the all arm's reward $i \in C_j$ are observed with posterior distributions updated simultaneously. Define $L_i(T) = \frac{ln(T)}{d(\mu_i, \mu_1)} = (1+\epsilon)^2 \frac{ln(T)}{d(\mu_i, \mu_1)}$, and let τ be the largest time step until $k_{C_j}(t) \leq L_i(T)$,

$$\begin{aligned} &\sum_{t=1}^{T} Pr(I(t) = i, \hat{\mu}(t) \leq x_i, \theta_i(t) \geq y_i) \\ \leq &\sum_{t=1}^{T} Pr(I(t) = i, \theta_i(t) \geq y_i | \hat{\mu}(t) \leq x_i) \\ \overset{(a)}{\leq} &L_i(T) + \sum_{t=\tau+1}^{T} e^{-td(\mu_i, \mu_1)} \\ \overset{(b)}{\leq} &L_i(T) + \sum_{t=\tau+1}^{T} \frac{1}{T} \\ \leq &L_i(T) + 1. \end{aligned} \tag{17}$$

Inequality (a) and (b) follow the Lemma 3 in [2] (based on the Chernoff-Hoeffding bounds). For $k_{C_j}(t) > L_i(T)$, $Pr(I(t) = i, \theta_i(t) \geq y_i | \hat{\mu}(t) \leq x_i) \leq \frac{1}{T}$.

Event 3:

$$\sum_{t=1}^{T} Pr(I(t) = i, \hat{\mu}(t) \leq x_i, \theta_i(t) \leq y_i) = O(1). \tag{18}$$

This proof also follows the Lemma 4 in [2].

Combining **Event 1-3**, we next derive the upper bound of **Case 1**. Similar to the algorithm 3, algorithm 4 updates all arms $i' \in C_j$ if the algorithm chooses $i \in C_j$. Let $M'_{C_j}(T) = \max_{i \in C_j} \{L_i(T)\}$ to ensure all arms have been pulled for sufficient times in cluster C_j, so that $\forall i \in C_j$ [2],

$$Pr(I(t) = i, \theta_i(t) \geq y_i | \hat{\mu}(t) \leq x_i) \leq \frac{1}{T}. \tag{19}$$

Let τ_{C_j} be the largest time step until $k_{C_j}(t) \leq M_{C_j}^{'}(T)$,

$$
\begin{aligned}
&E\left[\sum_{i\in C_j}\sum_{t=1}^{T}\mathbb{1}_{I(t)=i}\right] \\
&=\sum_{i\in C_j}\sum_{t=1}^{T}[Pr(I(t)=i,\hat{\mu}(t)>x_i) \\
&\qquad + Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\geq y_i) \\
&\qquad + Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\leq y_i)] \\
&=\sum_{i\in C_j}\sum_{t=1}^{T}[Pr(I(t)=i,\hat{\mu}(t)>x_i) \\
&\qquad + Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\leq y_i)] \\
&\qquad + \sum_{t=1}^{\tau_{C_j}}\sum_{i\in C_j} Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\geq y_i) \\
&\qquad + \sum_{i\in C_j}\sum_{t=\tau_{C_j}+1}^{T} Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\geq y_i) \\
&\overset{(a)}{\leq} \sum_{i\in C_j}\left[\frac{1}{d(x_i,\mu_i)}+1+1+1\right] + M_{C_j}^{'}(T) \leq M_{C_j}^{'}(T) + O(\frac{|C_j|}{\epsilon^2}).
\end{aligned}
\tag{20}
$$

Inequality (a) is because the algorithm only chooses one arm at each time slot, so the summation $\sum_{t=1}^{\tau_{C_j}}\sum_{i\in C_j} Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\geq y_i) \leq M_{C_j}^{'}(T)$.

□

*2) **Case 2**: $i \in C_1, i \neq 1$:* The upper bound of the cumulative regret of Case 2 follows the next lemma:

Lemma 4. *The cumulative regret within the cluster C_1 is upper bounded by*

$$\Delta_{C_1} M_{C_1}^{'}(T) + O(\frac{|C_1|-1}{\epsilon^2}), \tag{21}$$

where $M_{C_1}^{'}(T) = \max_{i\in C_1, i\neq 1}\left\{(1+\epsilon)^2\frac{ln(T)}{d(\mu_i,\mu_1)}\right\}$.

Proof.

$$
\begin{aligned}
&E\left[\sum_{i\in C_1, i\neq 1}\sum_{t=1}^{T}\mathbb{1}(I(t)=i)\right] \\
&=\sum_{i\in C_1, i\neq 1}\sum_{t=1}^{T}\{Pr(I(t)=i,\hat{\mu}(t)>x_i) \\
&\qquad + Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\geq y_i) \\
&\qquad + Pr(I(t)=i,\hat{\mu}(t)\leq x_i,\theta_i(t)\leq y_i)\} \\
&\overset{(a)}{\leq} \sum_{i\in C_1, i\neq 1}\left\{[1+\frac{1}{d(x_i,\mu_i)}]+1+1\right\} + M_{C_1}^{'}(T) \\
&\leq M_{C_1}^{'}(T) + O(\frac{|C_1|-1}{\epsilon^2}).
\end{aligned}
\tag{22}
$$

Inequality (a) follows the result from **case 1**. □

Combining the upper bounds of **Case 1** and **Case 2**, we get the upper bound for TS under clustered feedback setting, which corresponds to the following Theorem.

Theorem 2. *For the N_C-cluster stochastic bandit problem, if we follow the Thompson sampling procedure given in Algorithm 4, the expected regret is:*

$$E[R(T)] \leq \sum_{j=1}^{N_C}\Delta_{C_j} M_{C_j}^{'}(T) + O(\frac{N}{\epsilon^2}), \tag{23}$$

where $M_{C_j}^{'}(T) = \max_{i\in C_j}\left\{(1+\epsilon)^2\frac{ln(T)}{d(\mu_i,\mu_1)}\right\}$.

Remark 2. *This result shows that the regret bound depends on the number of clusters N_C, this feedback setting can be viewed as general because, when $L = 1$, it becomes bandit information feedback setting. The regret bound congruous with the results from [2] under the same setting. When $L = N$, it becomes the full information feedback setting. The regret bound becomes:*

$$E[R(T)] \leq \Delta_{\max} M^{'}(T) + O(\frac{N}{\epsilon^2}), \tag{24}$$

where $M^{'}(T) = \max_{i\in\{1,2,\ldots,N\}}\left\{(1+\epsilon)^2\frac{ln(T)}{d(\mu_i,\mu_1)}\right\}$, *and* $\Delta_{\max} = \max_{i\in\{1,2,\ldots,N\}}(\mu_1-\mu_i)$. *This result shows that the regret bound does not depend on the number of arms, because no matter which action the decision-maker selects, the rewards of all actions are returned and the parameters for these arms are updated simultaneously.*

V. NUMERICAL ANALYSIS

In this section, we simulate to validate our analysis. In the first experiment, we compare the algorithms of Thompson sampling and UCB_1 under different feedback settings with Bernoulli rewards. In the second experiment, we further apply these algorithms to a real-world dataset and compare the performance of each algorithm under different settings.

A. Bernoulli Settings

We first use binary rewards to compare with UCB_1 and Thompson sampling. We first randomly generate $K = 15$ arms with mean values uniformly chosen from [0.1,0.9]. In the simulation, once an arm i is played, a random reward is drawn from a Bernoulli distribution according to its mean value and independent of previous rewards. For the clustered feedback setting, we divide arms into N_C clusters randomly ($N_C = 3, 5$), and each cluster C_j contains L arms ($L = 5, 3$). We denote the total number of plays $T = 5000$, and compare the cumulative regret of different algorithms. Each algorithm runs 10 times, and the results are shown in Figure 2. The x-axis represents the time slot, and the y-axis represents the cumulative regret. From the simulation result, we can see that the algorithms in full information feedback setting ($L = 15$) outperform the clustered information feedback setting ($L = 3, 5$) and bandit feedback setting ($L = 1$). Meanwhile, in the clustered information setting, $L = 5$ has better performance than $L = 3$ scenario. The reason is that the expected regret is depended on the number of cluster N_C. $L = 5$ has $N_C = 3$ clusters, and $L = 3$ has $N_C = 5$ clusters. These results conform to the regret bound in our analysis.

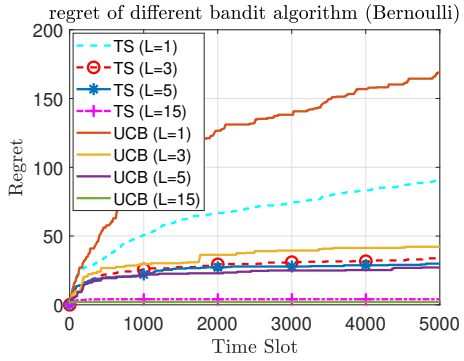

Fig. 2. Comparison of full information setting and clustered feedback setting with Bernoulli rewards.

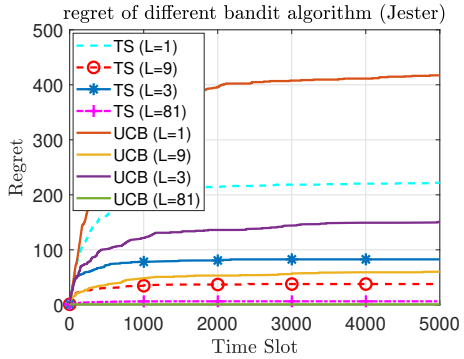

Fig. 3. Comparison of full information setting and clustered information setting with jester dataset.

B. Jester dataset

In the second experiment, we apply the algorithms to a subset of jester dataset [9]. The original dataset includes 2.3 million continuous ratings of 81 jokes from 25K users (between April 1999 - May 2003). Similar to the first simulation, we divide arms into N_C clusters randomly ($N_C = 9.27$). Each user rated several jokes and each rating is a real number between -10.00 and 10.00 (We can normalize the rating between -1.00 and 1.00). From Figure 3, we can see that algorithms with full information feedback ($L = 81$) outperform the ones under clustered information feedback settings ($L = 3, 9$). We also observe that the Thompson sampling algorithm has a better performance than the UCB_1 algorithm. The cumulative regret of the full information feedback setting is closer to the $\frac{1}{81}$ of the bandit feedback setting, and $L = 9$ has better performance than the $L = 3$ scenario. It is an expected result since the full information setting does not depend on the number of arms and the clustered information setting depends on the number of clusters N_C.

VI. CONCLUSION

In this paper, we study the MAB algorithms under the clustered information feedback setting, where the rewards of a cluster of arms are observable in each round. We upper bound the problem-dependent cumulative regret for UCB_1 and Thompson sampling algorithms under this setting. Theoretical analysis shows that the cumulative regret depends on the number of cluster N_C. Simulation and experiment results validate our theoretical analysis.

In terms of further works, a possible direction is to analyze the regret bound of another bandit algorithm (e.g. KL-UCB). As the upper bounds of the MAB problems are typically very loose, next, we will analyze the lower bound of each algorithm in the clustered information feedback setting.

REFERENCES

[1] Shipra Agrawal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[2] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial Intelligence and Statistics*, pages 99–107, 2013.

[3] Peter Auer. Using confidence bounds for exploitation-exploration tradeoffs. *J. Mach. Learn. Res.*, 3(null):397–422, March 2003.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[5] Hildo Bijl, Thomas B Schön, Jan-Willem van Wingerden, and Michel Verhaegen. A sequential monte carlo approach to thompson sampling for bayesian optimization. *arXiv preprint arXiv:1604.00169*, 2016.

[6] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721, 2012.

[7] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, Gilles Stoltz, et al. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

[8] Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376, 2011.

[9] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

[10] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142, 2018.

[11] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *Artificial intelligence and statistics*, pages 592–600, 2012.

[12] Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251, 2013.

[13] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[14] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[15] Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

[16] Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Found. Trends Mach. Learn.*, 11(1):1–96, July 2018.

[17] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

[18] Yingce Xia, Tao Qin, Weidong Ma, Nenghai Yu, and Tie-Yan Liu. Budgeted multi-armed bandits with multiple plays.

[19] Tianchi Zhao, Ming Li, and Matthias Poloczek. Fast reconfigurable antenna state selection with hierarchical thompson sampling. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.