Censored Semi-Bandits for Adaptive Resource Allocation

Abstract

We consider the problem of sequentially allocating resources in a *censored semi-bandits* setup, recently introduced in Verma et al. (2019). Here, the learner allocates resources at each step, receives censored feedback for the same, and the goal is the learn *good* allocations over time. The feedback at each step depends on two hidden parameters, one specific to the arm but independent of the allocation and one that depends on the allocation. Verma et al. (2019) proposed a two-step algorithm for this problem, where the two sets of problem parameters were estimated in separate steps, leading to potential loss of optimality. The problem of developing algorithms that jointly estimate the parameters was left open. In this work, we develop novel sequential resource allocation algorithms that not only close this open problem but also extend their applicability to several natural settings where the parameters of interest could be clustered. Our algorithms are better and more natural in the sense that they are the time horizon independent, do not rely on extraneous inputs that were previously needed as stopping criteria, and can adapt well to cases where the arms are clustered. We prove regret guarantees for our algorithms and experimentally demonstrate their improved performance with respect to state of the art.

1 Introduction

Sequential allocation problems with a censored feedback structure have received significant interest in recent times. Censoring occurs naturally in several applications such as police patrolling of opportunistic crimes (Curtin et al., 2010), poaching control (Nguyen et al., 2016; Gholami et al., 2018), traffic enforcement (Adler et al., 2014; Rosenfeld and Kraus, 2017), supplier selection (Abernethy et al., 2016), budget allocation (Lattimore et al., 2014, 2015; Dagan and Koby, 2018) and several others. In all these applications, an allocation to a set of options (allocation of price to items, allocation of the patrol to locations, etc.,) is made by a learner, and then the response to the allocation (purchase, crime, etc.) is observed. If a positive response is observed (crime occurs, a purchase is made), the learner receives information about the goodness of allocation. However, if a negative response (censored) is observed (no crime, no purchase, etc.), the learner cannot decide if it is because of a good allocation or because of an inherent low propensity of a response. The goal then is to make repeated allocations and learn the *best* possible allocation strategy.

Classical approaches to this problem learn from historical data Curtin et al. (2010); Adler et al. (2014); Zhang et al. (2016); Rosenfeld and Kraus (2017). Game theoretic approaches have also been considered Nguyen et al. (2016); Gholami et al. (2018); Sinha et al. (2018) where the user (criminal, buyer, etc.) knows the history of allocations and responds strategically. We consider a more realistic scenario where the user responds *opportunistically*, i.e., the user's response is stochastic and based only on the current allocation. This setup was introduced in Verma et al. (2019), where a two-stage combinatorial semi-bandits algorithm was proposed under a natural behavioral model of the users. Under this model, when presented with an allocation, a user's response has a generative structure. First, the user tosses a coin with a certain bias to decide to respond to the allocation. If the decision is a yes, then the user responds if the allocation is below a certain user-specific hidden threshold. For instance, in the case of crime patrol allocation, a criminal first probabilistically *decides* to commit a crime at a location and *proceeds* to commit it if the patrol allocation at the location is below the criminal specific crime threshold. While the setup is natural and covers a broad range of applications, the algorithm proposed in Verma et al. (2019) has several fundamental shortcomings that we address in this work.

Firstly, the algorithm proposed by Verma et al. (2019) has a two-step procedure where the parameters of the user's generative model (the bias to commit a crime, the crime-threshold) are estimated separately in each step. As it is clear, this may lead to a loss of potential learning

Preliminary work. Under review by AISTATS 2020. Do not distribute.

opportunity of the parameters of the second step (the crime-threshold) during the process of learning the first step. More importantly, the algorithm of Verma et al. (2019) requires as input an accuracy tolerance parameter that decides the stopping criteria for the estimation of the first step. It is not clear how to decide this parameter in practice. Furthermore, the algorithm proposed is time horizon dependent, whereas most bandit algorithms are independent of the same. Lastly, while Verma et al. (2019) discuss two cases of crime-thresholds, one where it is the same for all users and other where it is different for different users, they do not discuss a natural setting where the users are clustered w.r.t the thresholds. For instance, consider users to have thresholds that are either *low, medium, or high*. This is a very natural setting in practical applications. However, the algorithm of Verma et al. (2019) will treat all these thresholds as different from each other.

In this work, we significantly improve state of the art for the problem of sequential allocation under a censored feedback structure. More precisely, we make the following contributions:

- We develop algorithm **CSB-JS** for the case where the threshold is the same for all arms in Section 3. In Section 4, we consider general case where arms can have different threshold and develop algorithm named **CSB-JD**.
- **CSB-JD** can be specialized to the case where the number of thresholds is smaller than the number of arms. We prove that the regret bound of algorithm depends on the number of unique thresholds.
- Our algorithms do not need time horizon T and ϵ (minimum mean loss) as input, unlike the state-of-art algorithms (Verma et al., 2019).
- Our algorithms have a more natural structure where we jointly estimate the unknown threshold and mean losses of arms (using a version of Thompson Sampling algorithms for Multiple Play Multi-Armed Bandits and Combinatorial Bandits) in contrast to algorithms in Verma et al. (2019) which have a two-step procedure.
- We experimentally show the advantage of using our algorithm over the state of the art sequential allocation algorithms (Section 5).

Related Work:

Several directions have been considered in resource allocation problems to tackle crime Curtin et al. (2010) ,Nguyen et al. (2016); Gholami et al. (2018) some of which learn from historical data while others are game theoretic. We work in an explore-exploit framework which is different from both these settings. (Badanidiyuru et al., 2018), (Abernethy et al., 2016; Jain and Jamieson, 2018), (Lattimore et al., 2014, 2015; Dagan and Koby, 2018) study similar setup but differ in the exact model that we work with. Our work is most related to Verma et al. (2019). Allocation problems for the combinatorial setting have been explored in (Cesa-Bianchi and Lugosi, 2012; Chen et al., 2013; Rajkumar and Agarwal, 2014; Combes et al., 2015; Chen et al., 2016; Wang and Chen, 2018) which again differ from our specific setting. Resource allocation with semi-bandits feedback (Lattimore et al., 2014, 2015; Dagan and Koby, 2018) study a related but less general setup where the reward linearly depends on allocation and a hidden threshold. Recently, this work is extended to concave reward function by Fontaine et al. (2019). Our setting requires an additional unknown parameter for each arm, a 'mean loss,' which also affects the reward. Our work is mostly related to that of Verma et al. (2019). While we work with the same behavioral model as Verma et al. (2019), we develop significantly improved algorithms with guarantees that have improved and more natural dependence on problem parameters.

2 Problem Setting

We consider a sequential learning problem where K denotes the number of arms (locations), and Q denotes the amount of divisible resources. We follow the setting of Censored Semi-Bandits (CSB) similar to that of Verma et al. (2019). In a CSB, corresponding to each arm i, we have a Bernoulli random variable with mean $\mu_i \in [0,1]$, whose realization in the t^{th} round is denoted by $X_{t,i}$. Each arm may be assigned a fraction of resource, which determines the feedback observed and the loss incurred from that arm. Formally, denoting the resources allocated to the arms by $\boldsymbol{a} := \{a_i : i \in [K] \mid a_i \in [0,Q]\}$, the loss incurred equals the realization of the arm $X_{t,i}$ if $a_i < \theta_i$, where $\theta_i \in [0,Q]$ is a fixed but unknown threshold. When $\theta_i \leq a_i$, which corresponds to the scenario when the allocated resources are more than the threshold factor, we do not observe $X_{t,i}$, and hence the loss equals 0. The vectors $\boldsymbol{\theta} := \{\theta_i\}_{i\in[K]}$ and $\boldsymbol{\mu} := \{\mu_j\}_{i\in[K]}$ are unknown and identify an instance of CSB problem, which we denote henceforth using $P = (\boldsymbol{\mu}, \boldsymbol{\theta}, Q) \in [0,1]^K \times \mathbb{R}_+^K \times \mathbb{R}_+$. The collection of all CSB instances is denoted as $\mathcal{P}_{\text{CSB}}$. For simplicity of discussion, we assume that means are ordered as $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_K$ and for any integer M, refer the first M arms in the order as the top-M arms. Of course, the algorithm is not aware of this order.

Optimal Allocation. If μ and θ are known, the goal is to arrive at an allocation that minimizes the total expected loss incurred. For instance $P \in \mathcal{P}_{\text{CSB}}$

when $\boldsymbol{a}$ and θ are known, the optimal allocation can be computed by solving the following 0-1 knapsack problem:

$$\boldsymbol{a}^\star \in \arg\min_{\boldsymbol{a}\in\mathcal{A}_Q} \sum_{i=1}^{K} \mu_i \mathbb{1}_{\{a_i<\theta_i\}}, \tag{1}$$

where $\mathcal{A}_Q = \{a_i \in [0,Q] | \sum_i a_i \leq Q\}$ denotes the set of all feasible resource allocations.

Parameter Estimation. Since μ and θ are unknown, we estimate the parameters μ and θ in an online fashion using the observations made at each round. In each round t, a random vector $\boldsymbol{X_t} = (X_{t,1}, X_{t,2}, \ldots, X_{t,K}) \in \{0,1\}^K$ is generated, where $\mathbb{E}[X_{t,i}] = \mu_i$ and the sequence $(X_{t,i})_{t\geq 1}$ is i.i.d. for all $i \in [K]$. Then a learner picks an allocation vector $\boldsymbol{a}_t \in \mathcal{A}_Q$ and observes a random feedback $\boldsymbol{Y_t} = \{Y_{t,i} : i \in [K]\}$, where $Y_{t,i} = X_{t,i}\mathbb{1}_{\{a_{t,i}<\theta_i\}}$ and incurs loss $\sum_{i\in[K]} Y_{t,i}$. We aim to design optimal strategies for online resource allocation such that the expected (pseudo) regret over a period of horizon T given by the following expression is minimized:

$$\mathbb{E}[\mathcal{R}_T] = \sum_{t=1}^{T}\sum_{i=1}^{K} \mu_i \left(\mathbb{1}_{\{a_{t,i}<\theta_i\}} - \mathbb{1}_{\{a_i^\star<\theta_i\}}\right).$$

Our goal is to learn a policy that gives sub-linear regret, i.e., $\mathbb{E}[\mathcal{R}_T]/T \to 0$ as $T \to \infty$

Drawbacks of existing solutions. Algorithms with sub-linear expected regret for CSBs have been studied in Verma et al. (2019)(CSB-ST for the same threshold case and CSB-DT for different threshold case). They studied the two settings (a) θ_i is same for all arms i, (b) θ_i takes different values. The algorithms they proposed to operate in two stages - (a) threshold estimation phase and (b) regret minimization phase. In addition, they require to know time horizon T and requires the mean losses to be larger than a certain threshold ($\epsilon > 0$). In this work, we attempt to remove these two requirements and propose anytime algorithms that do not require to know any instance-specific information.

2.1 Allocation Equivalent

Next, we define when a pair of thresholds for the given loss vector and resources to be 'equivalent.'

Definition 1 (Allocation Equivalent)**.** *For any loss vector $\boldsymbol{\mu}$ and fix resources Q, two threshold vectors $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are* allocation equivalent *iff the following holds:*

$$\min_{\boldsymbol{a}\in\mathcal{A}_Q} \sum_{i=1}^{K} \mu_i \mathbb{1}_{\{x_i\geq\theta_i\}} = \min_{\boldsymbol{a}\in\mathcal{A}_Q} \sum_{i=1}^{K} \mu_i \mathbb{1}_{\{x_i\geq\hat{\theta}_i\}}.$$

Such equivalence allows us to find the threshold vector within fix error tolerance, which has the same total mean loss reduction as a true threshold vector has.

3 Arms with Same Threshold

We first focus on the simple case where $\theta_i = \theta_s$ for all $i \in [K]$ to bring out the main ideas of the algorithm we develop. With abuse of notation, we continue to denote an instance of CSB with the same threshold as $(\boldsymbol{\mu}, \theta_s, Q)$ where $\theta_s \in (0, Q]$. Note that the threshold is the same, but the mean losses can be different across the arms.

Though θ_s can take any value in the interval $(0, Q]$, an allocation equivalent to θ_s can be confined to a finite set. The following lemma shows that an allocation equivalent lies in a set consisting of the K elements.

Lemma 1. *Let $\theta_s \in (0, Q]$, $M = \min\{\lfloor Q/\theta_s \rfloor, K\}$ and $\hat{\theta}_s = Q/M$. Then θ_s and $\hat{\theta}_s$ are allocation equivalent. Further, $\hat{\theta}_c \in \Theta$ where $\Theta = \{Q/K, Q/(K-1), \cdots, Q\}$.*

Proof. The proof is a straight forward adapted of Lemma 1 in Verma et al. (2019) by allowing the threshold to be any value in $(0, Q]$ where Q can be great than 1. As all arms have the same threshold, the learner has to equally divided resources among selected arms. There are such K choices where choice $k \in [K]$ means resources are equally divided among k arms. One of these choices is an allocation equivalent to θ_s. □

Once the threshold is known, the optimal policy of a learner is to allocate $\hat{\theta}_s$ fraction of resource among M arms having the highest mean loss. As initially, mean losses are not known, empirical estimates of the losses can be used. When resource $\hat{\theta}_s$ is allocated to M arm having the highest empirical loses, no loss is observed from them, but a loss of each of the remaining $K - M$ arms are observed (semi-bandits). In bandits literature, such a problem where once can sample rewards from a subset of arms is known as Stochastic Multiple-Play Multi-Armed Bandits (MP-MAB) problem. Thus once the learner identifies a threshold equivalent of θ, the CSB problem is equivalent to solving an MP-MAB problem (Verma et al., 2019). As MP-TS algorithm (Komiyama et al., 2015) is shown to achieve optimal regret bound for Bernoulli distributions, we adapt it to our problem.

3.1 Algorithm: CSB-JS

We develop a Thompson-sampling based algorithm named **CSB-JS** for the same threshold case that simultaneously estimates threshold and mean losses. **CSB-JS** starts with equally distributing the resources among all the K arms and continues to do the same in the following rounds until loss 0 is observed on any of the arms. Once loss 1 is observed on any of the arms,

then it equally distributes the resources among $K-1$ arms. The process is repeated till the end. Along the way, the algorithms identify the allocation equivalent of θ_s and also learns the optimal allocation of resources. This is in contrast with the CSB-ST algorithm in Verma et al. (2019), which does it two separate phases, one followed after the other.

The pseudo-code of the algorithm is given in Algorithm in **CSB-JS**. It works as follows: it takes Q and K as inputs. For all $i \in [K]$ we use variables S_i and F_i to keep track of the number of rounds in which we observe loss 1 and 0, respectively, where loss value is observed form arm i only when it receives at least θ_i amount of resource. The prior loss distribution of each arm is set as the Beta distribution $\beta(1,1)$ by initializing $S_i = 1$ and $F_i = 1$ (Line 2). For each arm $i \in [K]$, Let $S_i(t)$

CSB-JS Joint estimation with Same threshold

1: **Input:** Q, K
2: Set $L = K, S_i = 1, F_i = 1, Z_i = 0 \quad \forall i \in [K]$
3: **for** $t = 1, 2, \ldots$ **do**
4: $\quad \forall i \in [K] : \hat{\mu}_i(t) \leftarrow \beta(S_i, F_i)$
5: $\quad A_t \leftarrow$ set of L arms with largest estimates
6: $\quad \forall i \in A_t$: allocate $\frac{Q}{L}$ resources and observe $X_{t,i}$
7: $\quad$ **if** $X_{t,j} = 1$ for any $j \in A_t$ **then**
8: $\quad\quad$ Set $L = L - 1$. $\forall i \in A_t$: update $S_i = S_i + X_{t,i}, F_i = F_i + 1 - X_{t,i} + Z_i$. $\forall j \in [K]$: set $Z_j = 0$
9: $\quad$ **else**
10: $\quad\quad \forall i \in A_t$: update $Z_i = Z_i + 1$
11: $\quad$ **end if**
12: $\quad \forall i \in [K] \setminus A_t$: update $S_i = S_i + X_{t,i}$, $F_i = F_i + 1 - X_{t,i}$
13: **end for**

and $F_i(t)$ denote the values of S_i and F_i at the starting of round t. In every round t, for each arm $i \in [K]$ a sample $\hat{\mu}_i$ is drawn from $\beta(S_i(t), F_i(t))$ independent of everything else (Line 4). **CSB-JS** finds top-L arms having largest empirical mean loss (denoted as set A_t in line 5) and equally distributes the resources among the arms in set A_t (Line 6). Note that initially we set $L = K$ i.e., we assign Q/K to each arm.

If a loss 1 is observed on any arms in set A_t (Line 7), it implies that current value of $\hat{\theta}_s$ is an underestimate of θ_s. Hence L is decreased by 1 and then the success and failure counts are also updated as $S_i = S_i + X_{t,i}, F_i = F_i + 1 - X_{t,i} + Z_i$ for each arm $i \in A_t$, and for the all $j \in [K]$, Z_j is reset to 0 (Line 8). Variables $(Z_i : \forall i \in [K])$, keep track of how many times loss 0 is observed for arms in set A_t (Line 10) before loss 1 is observed on any of them. It is reset to zero for all arms once loss 1 is observed for any arm in A_t. Variable $Z_i, i \in [k]$ is useful to distinguish zeros observed on arm i when it receives under and over allocation of resource.

If no loss is observed for all arms in the set A_t, Z_i is incremented by 1 for each arm $i \in A_t$. The values of S_i and F_i are updated for each arm where no resources are allocated (Line 12).

Since Θ has finite size, the allocation equivalent of θ_s is found in the finite number of rounds. After this, the algorithm allocates resources equally among M arms (Lemma 1) in the subsequent rounds and observes loss samples for remaining $K-M$ arms. The selected arms correspond to top-M arms with the highest estimated mean losses that are obtained from Thompson Sampling (Line 4). Hence after an allocation equivalent of θ_s is reached, in each round samples from the $K-M$ arms are observed which corresponds to selecting the $K-M$ arms with the smallest means, i.e., **CSB-JS** is the same as MP-TS that plays $K-M$ arms in each rounds and aims to minimize the sum of mean losses incurred from $K-M$ arms. We exploit this observation to adapt the regret bounds of MP-TS.

3.2 Analysis of CSB-JS

Let T_{θ_s} denote number of rounds required to find an allocation equivalent of θ_s and and R_{θ_s} denote the regret incurred in this period. For instance $(\boldsymbol{\mu}, \theta_c, Q)$ and any feasible allocation $\boldsymbol{a} \in \mathcal{A}_Q$, we define $\Delta_a = \sum_{i=1}^{K} \mu_i \left(\mathbb{1}_{\{a_i < \theta_i\}} - \mathbb{1}_{\{a_i^* < \theta_i\}} \right)$ and $\Delta_m = \max_{\boldsymbol{a} \in \mathcal{A}_Q} \Delta_a$. The first result gives the lower and upper bounds on expected value of T_{θ_s}.

Lemma 2. *The expected number of rounds needed by* **CSB-JS** *to find an allocation equivalent of θ_s is bounded as*

$$\mathbb{E}\left[T_{\theta_s}\right] \geq \sum_{L=M+1}^{K} \frac{1}{1 - \Pi_{i \in [L]}(1-\mu_i)} \quad \text{and}$$

$$\mathbb{E}\left[T_{\theta_s}\right] \leq \sum_{L=M+1}^{K} \frac{1}{1 - \Pi_{i \in [K]/[K-L+1]}(1-\mu_i)}.$$

We are now ready the state the regret bounds.

Theorem 1. *Let $\mu_M > \mu_{M+1}$ and $T > T_{\theta_s}$. Then the regret of* **CSB-JS** *is upper bound as*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \sum_{L=M+1}^{K} \frac{\Delta_m}{1 - \Pi_{i \in [K]/[K-L+1]}(1-\mu_i)} + O\left(\sum_{i=1}^{M} \frac{(\mu_{M+1} - \mu_i) \log T}{d(\mu_i, \mu_{M+1})} \right).$$

Proofs of the above results appear in the supplementary. Proof of Lemma 2 follows by deriving the distribution of number of rounds required to observe sample 1 on any of given set of independent Bernoulli random variable. The first term in the regret bound in

Theorem 1 corresponds to the expected regret incurred while finding the allocation equivalent. The second tern in the regret bound corresponds to the expected regret due to the MB-MAB based regret minimization algorithm. The assumption $\mu_M > \mu_{M+1}$ ensures that Kullback-Leibler divergence in the bound is well defined. This assumption is also equivalent to saying that the set of top-M arms is unique.

Lower bound: As a consequence of the equivalence, a lower bound on MP-MAB is also a lower bound on CSB with the same threshold. Hence the following lower bound given for any strongly consistent algorithm (Anantharam et al., 1987, Theorem 3.1) is also a lower bound for the CSB problem:

$$\lim_{T\to\infty} \mathbb{P}\left\{\mathbb{E}[\mathcal{R}_T] \geq \sum_{i=1}^{M} \frac{(1-o(1))(\mu_{M+1}-\mu_i)\log T}{d(\mu_i,\mu_{M+1})}\right\} = 1$$

where $d(p,q)$ is the KL divergence between two Bernoulli distributions with parameter p and q.

Corollary 1. *The regret of* **CSB-JS** *is asymptotically optimal.*

The proof follows by comparing the asymptotic regret bound of **CSB-JS** with the above lower bound.

4 Arms with Different Threshold

We now consider a more general case where the threshold may not be the same for all arms. The first difficulty with this setup is to find an optimal allocation which need not be just allocating resource to top M arms. Verma et al. (2019) gives the following result to find the optimal allocation for an instance with different thresholds in $\mathcal{P}_{\text{CSB}}$. Let $KP(\boldsymbol{\mu},\boldsymbol{\theta},Q)$ denote a 0-1 knapsack problem with capacity Q and K items where item i has weight θ_i and value μ_i.

Proposition 1. *Let $P = (\boldsymbol{\mu},\boldsymbol{\theta},Q) \in \mathcal{P}_{CSB}$. Then the optimal allocation for P is a solution of $KP(\boldsymbol{\mu},\boldsymbol{\theta},Q)$.*

Assigning θ_i resource to arm i decreases the total mean loss by an amount μ_i. As the goal is to allocate resources such that the total mean loss is minimized, i.e., $\min_{\boldsymbol{a}\in\mathcal{A}_Q} \sum_{i\in[K]} \mu_i \mathbb{1}_{\{a_i<\theta_i\}}$. It is equivalent to solving a 0-1 knapsack with capacity Q where item i has weight θ_i and value μ_i.

Another difficulty of having different thresholds is that the estimation of the threshold for each arm is needed to be done separately. Unfortunately, we do not have a result equivalent of Lemma 1 so that the search space can be restricted to a finite set. We need to search over the entire $(0, Q]$ interval for each arm.

Let $r := Q - \sum_{i:a_i^\star \geq \theta_i} \theta_i$, where r is the residual resources after optimal allocation. Define $\gamma = r/K$. Any instance with $\gamma = 0$ becomes a 'hopeless' problem instance as the only vector that is allocation equivalent to $\boldsymbol{\theta}$ is $\boldsymbol{\theta}$ itself, i.e., $a_i^\star = \theta_i$, $\forall i \in [K]$, which needs θ_i values to be estimated accurately to achieve optimal allocation. However, for $\gamma > 0$, the allocation equivalent can be found with small errors in θ_i values hence hence achievable in finite time. The proof follows similar to Lemma 3 of Verma et al. (2019).

Lemma 3. *Let $\gamma = r/K$ and $\forall i \in [K] : \hat{\theta}_i \in [\theta_i, \lceil \theta_i/\gamma \rceil \gamma]$. Then $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are allocation equivalent.*

Once we estimate the threshold $\boldsymbol{\theta}$ with accuracy such that the estimate $\hat{\boldsymbol{\theta}}$ is an allocation equivalent of $\boldsymbol{\theta}$, the problem is equivalent to solving the $KP(\boldsymbol{\mu},\hat{\boldsymbol{\theta}},Q)$ provided we learn $\boldsymbol{\mu}$. The latter part is equivalent to solving a Combinatorial Semi-Bandits (Chen et al., 2013; Combes et al., 2015; Wang and Chen, 2018). Combinatorial Semi-Bandits is a generalization of MP-MAB, where one needs to identify a subset of arms (from a collection of subsets) such that sum of reward/loss of the arm is the subset is the highest/lowest and the size of the subset selected in each round need not be the same. We could use an algorithm that works well for the Combinatorial Semi-Bandits, like SDCB Chen et al. (2016) and CTS Wang and Chen (2018) for solving CSB problem. CTS uses Thompson Sampling, whereas SDCB uses the UCB type index.

4.1 Algorithm: CSB-JD

We develop an algorithm named **CSB-JD** for the case where we do not have prior information that the thresholds are all the same. It is an adaptation of Thompson Sampling based combinatorial semi-bandits algorithm CTS and exploits Lemma 3 to find allocation equivalent. **CSB-JD** works as follows: it takes Q, K, and γ as input. We initialize the prior distribution of each arm as the Beta distribution $\beta(1,1)$, which is similar to **CSB-JS**. For each arm $i \in [K]$, algorithm maintains a variable L_i and set Z_i. L_i is the lower bound of the threshold for arm i, set Z_i keeps count of the number of time 0 on the arm i for different resource allocations, and $Z_i[a_i]$ represents the count of zeros for allocation a_i. L_i is initialized as 0 and Z_i as an empty set (Line 2). $Z_i[\cdot]$ plays a role similar to scalar Z_i in **CSB-JD**; however, it needs to store the counts for different allocations.

Let $S_i(t)$ and $F_i(t)$ denote the value of S_i and F_i at beginning of the round t. In round t, for each $i \in [K]$ an independent sample of $\hat{\mu}_{t,i}$ is drawn from $\beta(S_i(t), F_i(t))$ (Line 4). Initially, the lower bound of the threshold for each arm is 0, and its value will be increased for an arm if loss 1 is observed. At the start, the resources are equally distributed among the arms. In the following rounds, it is incremented by an amount of γ for arms

CSB-JD Joint estimation with Different threshold

1: **Input:** Q, K, γ
2: $\forall i \in [K]$: set $S_i = 1, F_i = 1, L_i = 0$, and $Z_i = \phi$
3: **for** $t = 1, 2, \ldots$ **do**
4: $\quad \forall i \in [K] : \hat{\mu}_{t,i} \leftarrow \beta(S_i, F_i)$.
5: $\quad$ **if** $Q - \sum_{i \in [K]: L_i \neq 0} (L_i + \gamma) \geq 0$ **then**
6: $\quad\quad$ Compute $\boldsymbol{a}_t$ using (2) and $A_t \leftarrow [K]$
7: $\quad$ **else**
8: $\quad\quad$ $A_t \leftarrow KP(\hat{\boldsymbol{\mu}}_t, \boldsymbol{a}_t, Q)$ where $a_{t,i} = L_i + \gamma$
9: $\quad$ **end if**
10: $\quad$ **for** $i \in A_t$ **do**
11: $\quad\quad$ Assign $a_{t,i}$ resources to arm i. Observe $X_{t,i}$
12: $\quad\quad$ **if** $X_{t,i} = 1$ **then**
13: $\quad\quad\quad$ If $L_i < a_{t,i}$ then change $L_i = a_{t,i}$
14: $\quad\quad\quad$ Set $S_i \leftarrow S_i + 1$, $F_i \leftarrow F_i + \sum_{a_i \leq L_i} Z_i[a_i]$
15: $\quad\quad\quad$ $\forall a_i \leq L_i$: set $Z_i[a_i] = 0$
16: $\quad\quad$ **else**
17: $\quad\quad\quad$ **if** $a_{t,i}$ is not in Z_i **then**
18: $\quad\quad\quad\quad$ Add $Z_i[a_{t,i}] = 1$ to set Z_i
19: $\quad\quad\quad$ **else**
20: $\quad\quad\quad\quad$ Update $Z_i[a_{t,i}] = Z_i[a_{t,i}] + 1$
21: $\quad\quad\quad$ **end if**
22: $\quad\quad$ **end if**
23: $\quad$ **end for**
24: $\quad \forall i \in [K] \setminus A_t : S_i = S_i + X_{t,i}, F_i = F_i + 1 - X_{t,i}$
25: **end for**

on which loss 1 is observed while uniformly distributing residual resources to others arms as follows:

$$a_{t,i} = \begin{cases} L_i + \gamma & \text{if } L_i \neq 0 \\ Q_l / L_0 & \text{otherwise} \end{cases} \tag{2}$$

where $Q_l := Q - \sum_{i \in [K]: L_i \neq 0} (L_i + \gamma)$ are the leftover resources and $L_0 := |\{i : L_i = 0\}|$ is the number of arms with 0 as their lower bound of threshold (Line 5-6). This approach gets a initial lower bound on thresholds for arms by exploiting all available resources. This is continued till each can be gets resource (Line 5-7).

When resources are not enough for all arms, then the set of arms is selected by solving $KP(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\theta}}_t, Q)$ problem (denoted as set A_t in Line 8). Each arm $i \in A_t$ has given resource $a_{t,i} = L_i + \gamma$ and a sample $X_{t,i}$ is observed (Line 11). If a loss is observed (Line 12) that implies the arm is under allocated and accordingly lower bound of threshold for arm is updated (Line 13). The success and failure counts are also updated as $S_i = S_i + 1, F_i = F_i + \sum_{a_i \leq L_i} Z_i[a_i]$ (Line 14), and values of set $Z_i[a_i]$ with $a_i \leq L_i$ are changed to 0 (Line 15).

If no loss is observed for arms having desired resource and $Z_i[a_i]$ is not in set Z_i then add $Z_i[a_i]$ to set Z_i with value 1 (Line 17-18) otherwise increment $Z_i[a_i]$ by 1 (Line 20). The success and failure counts are also updated for each arm $i \in [K] \setminus A_t$ as $S_i = S_i + X_{t,i}$ and $F_i = F_i + 1 - X_{t,i}$ (Line 24).

Remark. The lower bound for an arm having zero mean loss remains 0 irrespective of its associated threshold. Therefore, when resources are not enough, **CSB-JD** allocate only γ amount of resources to such arms.

4.2 Analysis of CSB-JD

Let T_{θ_d} denote number of rounds required to find an allocation equivalent for problem instance $(\boldsymbol{\mu}, \boldsymbol{\theta}, Q)$. An upper bound on expected value of T_{θ_d} is given by our next result.

Lemma 4. *Let $\gamma > 0$. Then, **CSB-JD** needs T_{θ_d} rounds to find allocation equivalent where*

$$\mathbb{E}[T_{\theta_d}] \leq \sum_{i \in [K]: \mu_i \neq 0} \left\lfloor \frac{\theta_i}{\gamma} \right\rfloor \left(\frac{1}{\mu_i} \right).$$

For instance $(\boldsymbol{\mu}, \theta, Q)$ and any feasible allocation $\boldsymbol{a} \in \mathcal{A}_Q$, we define $\Delta_a = \sum_{i=1}^{K} \mu_i (\mathbb{1}_{\{a_i < \theta_i\}} - \mathbb{1}_{\{a_i^\star < \theta_i\}})$, and $\Delta_m = \max_{\boldsymbol{a} \in \mathcal{A}_Q} \Delta_a$. We are now ready the state the regret bounds.

Theorem 2. *Let $\gamma > 0$, $S_a = \{i : a_i < \theta_i\}$ for any feasible allocation $\boldsymbol{a}$, and $k_\star = |S_{a^\star}|$. Then for any η such that $\forall \boldsymbol{a} \in \mathcal{A}_Q, \Delta_a > 2(k_\star^2 + 2)\eta$, the expected regret of **CSB-JD** is upper bound as*

$$\mathbb{E}[\mathcal{R}_T] \leq \sum_{i \in [K]: \mu_i \neq 0} \left\lfloor \frac{\theta_i}{\gamma} \right\rfloor \left(\frac{1}{\mu_i} \right) \Delta_m + O\left(\sum_{i=1}^{K} \max_{S_a : i \in S_a} \frac{8 |S_a| \log T}{\Delta_a - 2(k_\star^2 + 2)\eta} \right).$$

The first term of expected regret is due to the regret incurred while finding the allocation equivalent. The expected number of rounds needed to find the allocation equivalent is given by Lemma 4. As Δ_m is the maximum regret can be incurred in any round, the upper bound on expected regret while finding allocation equivalent is bounded by $\Delta_m \mathbb{E}[T_{\theta_d}]$. The second term corresponds to the expected regret due to the combinatorial semi-bandits algorithm.

4.3 Arms with n Different Threshold Classes

The following definition describes when we can say that two thresholds are different.

Definition 2 (Different Threshold)**.** *Two thresholds θ_i and θ_j are* different *if $\lceil \theta_i / \gamma \rceil \neq \lceil \theta_j / \gamma \rceil$.*

Using Lemma 3 and the above definition implies that two thresholds are different if they have different

Cases \ $\mathbb{E}$ [Rounds]	Verma et al. (2019) (requires to know a lower bound on μ_is (ϵ) and confidence parameter (δ))	This paper (does not requires to know ϵ and δ)
Same Threshold	$\frac{\log(\log_2(\lvert\Theta\rvert)/\delta)}{\log(1/(1-\epsilon))}\log_2(\lvert\Theta\rvert)$	$\sum_{L=M+1}^{K}\frac{1}{1-\Pi_{i\in[K]/[K-L+1]}(1-\mu_i)}$
Different Threshold	$\frac{K\log(K\log_2(\lceil 1+1/\gamma\rceil)/\delta)}{\log(1/1-\epsilon)}\log_2(\lceil 1+1/\gamma\rceil)$	$\sum_{i\in[K]:\mu_i\neq 0}\left\lfloor\frac{\theta_i}{\gamma}\right\rfloor\left(\frac{1}{\mu_i}\right)$
$1<n=\#$ Thresholds $<K$	-	$\sum_{i\in A_{\theta_n}}\left\lfloor\frac{\theta_i}{\gamma}\right\rfloor\left(\frac{1}{\mu_i}\right)+\sum_{k\in A^c_{\theta_n}:\mu_k\neq 0}\frac{n}{\mu_k}$

Table 1: Comparing upper bounds on the expected number of rounds needed to find allocation equivalent.

thresholds in the allocation equivalent vector. With abuse of notation, we denote an instance of CSB with n different thresholds as $(\boldsymbol{\mu}, \boldsymbol{\theta}_n, Q)$. Note that the optimal allocation for $(\boldsymbol{\mu}, \boldsymbol{\theta}_n, Q)$ is also a solution of $KP(\boldsymbol{\mu}, \boldsymbol{\theta}_n, Q)$ problem.

Let n be the number of different thresholds and G_i be the set of arms having the same threshold $\hat{\theta}_i$ in the allocation equivalent vector. Define $A_{\theta_n} := \left\{\arg\min_{j\in G_i:\mu_j\neq 0}\mu_j : \forall i \in [n]\right\}$ and $A^c_{\theta_n}$ represents the remaining arms. A_{θ_n} is the set of n arms having different threshold and minimum positive mean value in their corresponding set G_i.

Let T_{θ_n} denote number of rounds required to find an allocation equivalent for instance $(\boldsymbol{\mu}, \boldsymbol{\theta}_n, Q)$. An upper bound on expected value of T_{θ_n} is given below.

Lemma 5. *Let $\gamma > 0$, n be the number of different thresholds. Then, any algorithm needs T_{θ_n} rounds to find allocation equivalent where*

$$\mathbb{E}\left[T_{\theta_n}\right] \leq \sum_{i\in A_{\theta_n}}\left\lfloor\frac{\theta_i}{\gamma}\right\rfloor\left(\frac{1}{\mu_i}\right)+\sum_{k\in A^c_{\theta_n}:\mu_k\neq 0}\frac{n}{\mu_k}.$$

We are now ready the state the regret bounds.

Theorem 3. *Let $\gamma > 0$, n be the number of different thresholds, $S_a = \{i : a_i < \theta_i\}$ for any feasible allocation $\boldsymbol{a}$, and $k_\star = |S_{a^\star}|$. Then for any η such that $\forall \boldsymbol{a} \in \mathcal{A}_Q, \Delta_a > 2(k_\star^2+2)\eta$, the expected regret of any algorithm is upper bound as*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \left(\sum_{i\in A_{\theta_n}}\left\lfloor\frac{\theta_i}{\gamma}\right\rfloor\left(\frac{1}{\mu_i}\right)+\sum_{k\in A^c_{\theta_n}:\mu_k\neq 0}\frac{n}{\mu_k}\right)\Delta_m + O\left(\sum_{i=1}^{K}\max_{S_a:i\in S_a}\frac{8|S_a|\log T}{\Delta_a - 2(k_\star^2+2)\eta}\right).$$

The proof is Similar to the Theorem 2 except the first term of expected regret is due to the regret incurred while finding the allocation equivalent with n different thresholds. It is upper bounded by $\Delta_m\mathbb{E}\left[T_{\theta_n}\right]$. The second term is same as Theorem 2.

Table 1 summarizes our bounds in comparison with the bounds obtained in Verma et al. (2019).

5 Experiments

We empirically evaluate the performance of our algorithms **CSB-JS** and **CSB-JD** on three synthetically generated instances. In the instances I and II, the threshold is the same for all arm, whereas, in instance III, thresholds vary across arms. Details of the instances are as follows:

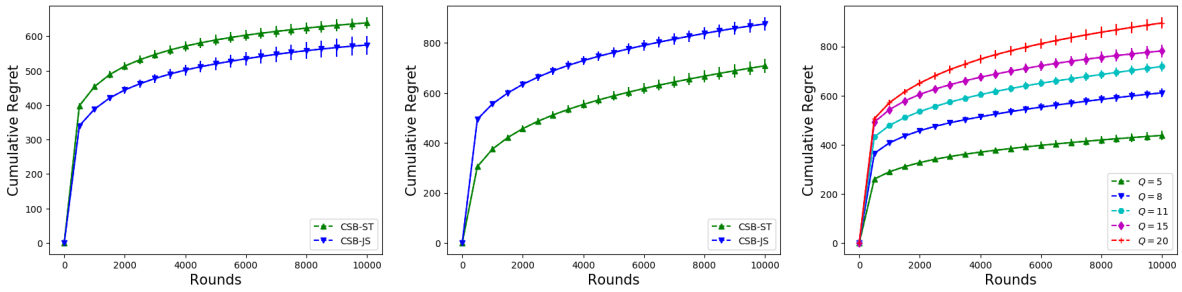
Identical Threshold: Both instance I and II have $K = 50, C = 20$, and $\theta_s = 0.7$. The mean loss of arm $i \in [K]$ is $x - (i-1)/100$. We set $x = 0.5$ for instance I and $x = 0.7$ for instance II.

Different Thresholds: Instance III has $K = 5$, and $\gamma = 10^{-2}$. The mean loss vector is $\boldsymbol{\mu} = [0.95, 0.9, 0.89, 0.33, 0.3]$ and the corresponding threshold vector is $\boldsymbol{\theta} = [0.8, 0.8, 0.8, 0.3, 0.3]$.

In all experiments, the losses of arm i are Bernoulli distributed with parameter μ_i. All experiments are repeated 50 times, and the regret curves are shown with a 95% confidence interval (the vertical line on each curve shows the confidence interval). We ran the **CSB-JS** for $T = 10000$ rounds in all the cases whereas $T = 5000$ for **CSB-JD**. The following empirical results validate sub-linear bounds for our algorithms.

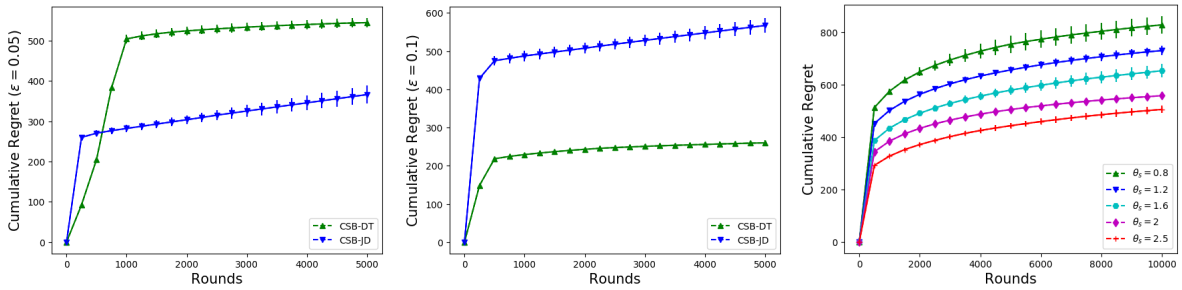
Experiments with the same threshold: We perform three different experiments on problem instances I and II. First, we compare **CSB-JS** with state-of-art CSB-ST algorithm (Verma et al., 2019) for CSB problem with same threshold. Even though our algorithm uses linear search to find allocation equivalent, it outperforms CSB-ST for instance I, where mean losses are small, as shown in (1a). As shown in (1b), CSB-ST performs better than our algorithm because it uses binary search and has to wait for less to observe losses when mean losses of the arm are large. Note that CSB-ST needs to know the lower bound on μ_i value, and it finds the allocation equivalent with a high probability $1-\delta$. We set lower bound on mean loss as $\epsilon = 0.1$ and confidence parameter $\delta = 0.1$.

Second, we vary the amount of resource Q while keeping other parameters are unchanged for instance II. With



(a) Comparison for Instance I (b) Comparison for Instance II (c) Varying Resources for II

Figure 1: Comparing cumulative regret of **CSB-JS** and CSB-ST (Verma et al., 2019) in (1a) and (1b). Comparison of cumulative regret and different amount of of resource for **CSB-JS** is shown in (1c).



(a) Comparison for Instance III (b) Comparison for Instance III (c) Varying Threshold for II

Figure 2: Comparing cumulative regret of **CSB-JD** and CSB-DT (Verma et al., 2019) in (2a) and (2b). Comparison of cumulative regret and different same threshold value for **CSB-JS** is shown in (2c).

more resources, the learner can allocate resources to more arms. Hence learner can observe losses from fewer arms in each round, which leads to slower learning and high cumulative regret, as shown in (1c). Third, we varied the threshold θ_s without changing other parameters for instance II. As a smaller threshold allows the allocation of resources to more arms, it leads to less feedback and slower learning. Hence regret increases as the threshold θ_s decreases, as shown in Fig. (2c).

Experiments with different thresholds: We compare **CSB-JD** on problem instances III with state-of-art CSB-DT algorithm (Verma et al., 2019) for CSB problem with different threshold. CSB-DT also needs to know the lower bound on μ_i value and confidence parameter. We set the confidence parameter $\delta = 0.01$. Similar to CSB-ST, CSB-DT also uses binary search to estimate the threshold for each arm. It uses the same threshold for fixed amount rounds, which depends upon the value of ϵ and δ. The smaller the value of ϵ, the more CSB-DT waits for observing a loss and incur more regret as well. On another hand, **CSB-JD** uses a linear search to estimate the threshold and does not need to know ϵ and δ. As shown in (2a), our algorithm outperforms CSB-DT for smaller ϵ value. Because for the smaller value of epsilon, CSB-DT has to wait for more rounds before changing to other thresholds without observing the loss hence incur more regret. For larger ϵ value, CSB-DT waits less and incur less regret compare to our algorithm, as shown in (2b).

6 Conclusion and Future Extensions

In this work, we studied the recently proposed Censored Semi-Bandits framework for adaptive resource allocation. While the existing algorithms for this problem separately estimated the parameters and required lower bounds on the mean loss as input, the algorithms discussed in this work are free of these inefficiencies. We also studied the setting when the thresholds are clustered into groups. We derive sub-linear regret guarantees for the proposed algorithms, and also show empirically that the proposed algorithms have better regret performance.

The settings considered so far do not make use of any metric similarities between the arms. For example, in the case of the police patrol allocation, the nearby nodes may have similar parameters, and we may be able to make use of the spatial coherence. Settings such as these are left for future work.

References

Arun Verma, Manjesh K Hanawal, Arun Rajkumar, and Raman Sankaran. Censored semi-bandits: A framework for resource allocation with censored feedback. *To appear in Advances In Neural Information Processing Systems*, 2019.

Kevin M Curtin, Karen Hayslett-McCall, and Fang Qiu. Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and Spatial Economics*, 10(1): 125–145, 2010.

Thanh H Nguyen, Arunesh Sinha, Shahrzad Gholami, et al. Capture: A new predictive anti-poaching tool for wildlife protection. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 767–775, 2016.

Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, , et al. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 823–831, 2018.

Nicole Adler, Alfred Shalom Hakkert, Jonathan Kornbluth, Tal Raviv, and Mali Sher. Location-allocation models for traffic police patrol vehicles on an interurban network. *Annals of Operations Research*, 221(1):9–31, 2014.

Ariel Rosenfeld and Sarit Kraus. When security games hit traffic: Optimal traffic enforcement under one sided uncertainty. In *IJCAI*, pages 3814–3822, 2017.

Jacob D Abernethy, Kareem Amin, and Ruihao Zhu. Threshold bandits, with and without censored feedback. In *Advances In Neural Information Processing Systems*, pages 4889–4897, 2016.

Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Optimal resource allocation with semi-bandit feedback. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 477–486. AUAI Press, 2014.

Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Linear multi-resource allocation with semi-bandit feedback. In *Advances in Neural Information Processing Systems*, pages 964–972, 2015.

Yuval Dagan and Crammer Koby. A better resource allocation algorithm with semi-bandit feedback. In *Proceedings of Algorithmic Learning Theory*, pages 268–320, 2018.

Chao Zhang, Victor Bucarey, Ayan Mukhopadhyay, Arunesh Sinha, Yundi Qian, Yevgeniy Vorobeychik, and Milind Tambe. Using abstractions to solve opportunistic crime security games at scale. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 196–204, 2016.

Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: Looking beyond a decade of success. In *IJCAI*, pages 5494–5501, 2018.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):13, 2018.

Lalit Jain and Kevin Jamieson. Firing bandits: Optimizing crowdfunding. In *International Conference on Machine Learning*, pages 2211–2219, 2018.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

Arun Rajkumar and Shivani Agarwal. Online decision-making in general combinatorial spaces. In *Advances in Neural Information Processing Systems*, pages 3482–3490, 2014.

Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015.

Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, pages 1659–1667, 2016.

Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 5101–5109, 2018.

Xavier Fontaine, Shie Mannor, and Vianney Perchet. A problem-adaptive algorithm for resource allocation. *arXiv preprint arXiv:1902.04376*, 2019.

Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pages 1152–1161, 2015.

V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays- part I. *IEEE Transactions on Automatic Control*, 32(11): 968–976, 1987.

Supplementary Material for 'Censored Semi-Bandits for Adaptive Resource Allocation'

A Proofs related to Section 3

Lemma 2. *The expected number of rounds needed by* **CSB-JS** *to find an allocation equivalent of* θ_s *is bounded as*

$$\mathbb{E}\left[T_{\theta_s}\right] \geq \sum_{L=M+1}^{K} \frac{1}{1-\Pi_{i\in[L]}(1-\mu_i)} \quad \textit{and} \quad \mathbb{E}\left[T_{\theta_s}\right] \leq \sum_{L=M+1}^{K} \frac{1}{1-\Pi_{i\in[K]/[K-L+1]}(1-\mu_i)}.$$

Proof. Let $X_1, X_2, \ldots, X_P$ be independent Bernoulli random variables where X_i has parameter μ_i. The samples from all random variables are observed at the same time. Let R_W is a random variable that counts the number of the rounds needed to observe loss 1 for any of $\{X_i\}_{i\in[P]}$. First, we compute $\mathbb{P}\{R_W = w\}$ i.e.,

$$\mathbb{P}\{R_W = w\} = \Pi_{i\in[P]}(1-\mu_i)^{w-1}\left(1-\Pi_{i\in[P]}(1-\mu_i)\right)$$

The previous results follows from the fact that there loss 1 is not observed for any of $\{X_i\}_{i\in[P]}$ in the first $w-1$ rounds and a loss is observed for at least one of the random variable in the w^{th} round. The expectation of R_W is given as follows:

$$\begin{aligned}
\mathbb{E}\left[R_W\right] &= \sum_{w=1}^{\infty} w\mathbb{P}\{R_W = w\} \\
&= \sum_{w=1}^{\infty} w\Pi_{i\in[P]}(1-\mu_i)^{w-1}\left(1-\Pi_{i\in[P]}(1-\mu_i)\right) \\
&= \left(1-\Pi_{i\in[P]}(1-\mu_i)\right)\sum_{w=1}^{\infty} w\Pi_{i\in[P]}(1-\mu_i)^{w-1}
\end{aligned}$$

Let $\bar{p} = 1-\Pi_{i\in[P]}(1-\mu_i)$, we have

$$\begin{aligned}
\implies \mathbb{E}\left[R_W\right] &= \bar{p}\sum_{w=1}^{\infty} w(1-\bar{p})^{w-1} \\
&= \bar{p}\left[\frac{d}{d\bar{p}}\sum_{w=1}^{\infty} -(1-\bar{p})^{w}\right] = \bar{p}\left[\frac{d}{d\bar{p}}\left(\frac{-1}{\bar{p}}\right)\right] \\
&= \bar{p}\left(\frac{1}{\bar{p}^2}\right) = \frac{1}{\bar{p}} \\
&= \frac{1}{1-\Pi_{i\in[P]}(1-\mu_i)}
\end{aligned}$$

Algorithm **CSB-JS** starts equal resources to all $L = K$ arm. When a loss 1 is observed for any of the arms, resources are equally allocated to $L = K-1$ arms. Let $T_\theta(L)$ denote the number of the rounds needed to see a loss 1 when L arms are allocated resources.

By taking top L arms in each round, the lower bound on expected value of $T_\theta(L)$ is given as:

$$\mathbb{E}\left[T_\theta(L)\right] \leq \frac{1}{1-\Pi_{i\in[L]}(1-\mu_i)}$$

Consider all wrong values of $L \in \{M+1, M+2, \ldots, K-1, K\}$, a lower bound on expected number of rounds needed to reach to correct allocation (*i.e.*, Q/M) is given as follows:

$$\mathbb{E}\left[T_{\theta_s}\right] = \sum_{L=M+1}^{K} \mathbb{E}\left[T_\theta(L)\right]$$

$$\implies \mathbb{E}\left[T_{\theta_s}\right] \geq \sum_{L=M+1}^{K} \frac{1}{1-\Pi_{i\in[L]}(1-\mu_i)}$$

Similarly taking bottom L arms in each round, the upper bound on the expected value of $T_\theta(L)$ is given as:

$$\mathbb{E}\left[T_\theta(L)\right] = \frac{1}{1-\Pi_{i\in[K]/[K-L+1]}(1-\mu_i)}$$

Consider all wrong values of $L \in \{M+1, M+2, \ldots, K-1, K\}$, the upper bound on expected number of rounds needed to reach to correct allocation (*i.e.*, Q/M) is given as follows:

$$\mathbb{E}\left[T_{\theta_s}\right] = \sum_{L=M+1}^{K} \mathbb{E}\left[T_\theta(L)\right]$$

$$\implies \mathbb{E}\left[T_{\theta_s}\right] \leq \sum_{L=M+1}^{K} \frac{1}{1-\Pi_{i\in[K]/[K-L+1]}(1-\mu_i)} \qquad \square$$

Now, we need the following results to prove the regret bound of **CSB-JS**.

Theorem 4. *Let $\hat{\theta}_s$ be allocation equivalent to θ_s for instance $(\boldsymbol{\mu}, \theta_s, Q)$. Then, the expected regret of the regret minimization phase of* **CSB-JS** *for T rounds is upper bound as*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq O\left((\log T)^{2/3}\right) + \sum_{i=1}^{M} \frac{(\mu_{M+1}-\mu_i)\log T}{d(\mu_i, \mu_{M+1})}. \tag{3}$$

Proof. The proof is adapted from Theorem 3 of Verma et al. (2019). As $\hat{\theta}_s$ be allocation equivalent to θ_s, the instances $(\boldsymbol{\mu}, \theta_s, Q)$ and $(\boldsymbol{\mu}, \hat{\theta}_s, Q)$ have same minimum value. Also, by the equivalence established by Verma et al. (2019), the regret minimization phase of **CSB-JS** is solving a MP-MAB instance. Then we can directly apply Theorem 1 of Komiyama et al. (2015) to obtain the regret bounds by setting $k = M$ and noting that we are in the loss setting and a mistake happens when a arm $i \in [M]$ is in selected superarm. $\square$

Theorem 1. *Let $\mu_M > \mu_{M+1}$ and $T > T_{\theta_s}$. Then the regret of* **CSB-JS** *is upper bound as*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \sum_{L=M+1}^{K} \frac{\Delta_m}{1-\Pi_{i\in[K]/[K-L+1]}(1-\mu_i)} + O\left(\sum_{i=1}^{M} \frac{(\mu_{M+1}-\mu_i)\log T}{d(\mu_i, \mu_{M+1})}\right).$$

Proof. The regret of **CSB-JS** consists of two parts: regret before knowing allocation equivalent and after knowing it. The expected regret incurred while estimating threshold equivalent is given by Lemma 2, which is the first part of regret. The second part of regret is due to the MP-MAB algorithm (MP-TS) and is given by Theorem 4. $\square$

B Proofs related to Section 4

Lemma 4. *Let $\gamma > 0$. Then,* **CSB-JD** *needs T_{θ_d} rounds to find allocation equivalent where*

$$\mathbb{E}\left[T_{\theta_d}\right] \leq \sum_{i\in[K]:\mu_i\neq 0} \left\lfloor \frac{\theta_i}{\gamma} \right\rfloor \left(\frac{1}{\mu_i}\right).$$

Proof. The expected number of rounds needed to observe a loss from an under-allocated arm with non-zero mean loss are $1/\mu_i$. When a loss is observed for an arm, **CSB-JD** increments resources by γ amount for that arm. In worse case, θ_i or more resources are allocated only after $\lfloor \theta_i/\gamma \rfloor$ number of increments in resource allocation for the arm i. Therefore, the expected number of rounds needed to estimate $\hat{\theta}_i \in [\theta_i, \lceil \theta_i/\gamma \rceil \gamma]$ are $\lfloor \theta_i/\gamma \rfloor (1/\mu_i)$.

Let consider the worst case where only one threshold is estimated at a time. Then the maximum expected rounds needed to estimate all thresholds are $\sum_{i\in[K]:\mu_i\neq 0} \lfloor \theta_i/\gamma \rfloor (1/\mu_i)$. With this argument, the proof is complete. $\square$

Remark: Note that **CSB-JD** can estimate thresholds of multiple arms by starting with the same allocation of resources to all arms. Hence the number of rounds needed for finding allocation equivalent might be very less than given by its upper bound in Lemma 4 where the worst case is considered.

We also need the following results to prove the regret bound of **CSB-JD**.

Theorem 5. *Let $\hat{\boldsymbol{\theta}}$ be allocation equivalent to $\boldsymbol{\theta}$ for instance $(\boldsymbol{\mu}, \boldsymbol{\theta}, Q)$, $S_a = \{i : a_i < \theta_i\}$ for any feasible allocation $\boldsymbol{a}$, $k_\star = |S_{a^\star}|$, and $M = \sum_{i\in[K]} \mathbb{1}_{\{a_i^\star \geq \theta_i\}}$. Then for any η such that $\forall \boldsymbol{a} \in \mathcal{A}_Q, \Delta_a > 2(k^{\star 2}+2)\eta$, the expected regret of the regret minimization phase of **CSB-JD** in T rounds is upper bound as $\left(\sum_{i\in[K]} \max_{S_a : i\in S_a} \frac{8|S_a|\log T}{\Delta_a - 2(k^{\star 2}+2)\eta}\right) + \left(\frac{KM^2}{\eta^2} + 3K\right)\Delta_m + \alpha_1 \left(\frac{8\Delta_m}{\eta^2}\left(\frac{4}{\eta^2}+1\right)^{k^\star} \log \frac{k^\star}{\eta^2}\right)$ where α_1 is a problem independent constant.*

The proof follows from Theorem 1 of Wang and Chen (2018) and Theorem 6 of Verma et al. (2019).

Theorem 2. *Let $\gamma > 0$, $S_a = \{i : a_i < \theta_i\}$ for any feasible allocation $\boldsymbol{a}$, and $k_\star = |S_{a^\star}|$. Then for any η such that $\forall \boldsymbol{a} \in \mathcal{A}_Q, \Delta_a > 2(k_\star^2 + 2)\eta$, the expected regret of **CSB-JD** is upper bound as*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \sum_{i\in[K]:\mu_i\neq 0} \left\lfloor \frac{\theta_i}{\gamma} \right\rfloor \left(\frac{1}{\mu_i}\right) \Delta_m + O\left(\sum_{i=1}^{K} \max_{S_a : i\in S_a} \frac{8|S_a|\log T}{\Delta_a - 2(k_\star^2+2)\eta}\right).$$

Proof. Similar to **CSB-JS**, the regret of **CSB-JD** also consists of two parts: regret before knowing allocation equivalent and after knowing it. We get the first part of expected regret by using the upper bound on the expected number of rounds needed to find allocation equivalent from Lemma 4. As Δ_m is the maximum regret can be incurred in any round. Once an allocation equivalent threshold is found, by exploiting equivalence with combinatorial semi-bandit, the second part of the expected regret is due to using a combinatorial semi-bandit algorithm (CTS) and is given by Theorem 5. □

Lemma 5. *Let $\gamma > 0$, n be the number of different thresholds. Then, any algorithm needs T_{θ_n} rounds to find allocation equivalent where*

$$\mathbb{E}\left[T_{\theta_n}\right] \leq \sum_{i\in A_{\theta_n}} \left\lfloor \frac{\theta_i}{\gamma} \right\rfloor \left(\frac{1}{\mu_i}\right) + \sum_{k\in A_{\theta_n}^c : \mu_k \neq 0} \frac{n}{\mu_k}.$$

Proof. As n is the number of different non-trivial equivalent thresholds, there are n different groups of arms where group G_i consists of arms having the same estimated threshold $\hat{\theta}_i$ in allocation equivalent. As we start with allocating γ resources among arms. If a loss is observed for the arms having desired resources, then resources are increased by γ for such arms in the next round.

The expected number of rounds needed to observe a loss by an under-allocated arm with mean loss $\mu_i > 0$ are $1/\mu_i$. Since resources are incremented by γ amount when a loss is observed for arm i, θ_i or more resources are allocated only after $\lfloor \theta_i/\gamma \rfloor$ number of increments in resource allocation to arm i. Therefore, the expected number of rounds needed to estimate $\hat{\theta}_i \in [\theta_i, \lceil \theta_i/\gamma \rceil \gamma]$ are $\lfloor \theta_i/\gamma \rfloor (1/\mu_i)$.

We divided the number of rounds to know allocation equivalent into two parts. The first deals with the maximum number of expected rounds needed to find n thresholds. The second part deals with finding the threshold for remaining arms using known n thresholds.

Let consider the worst case where only one threshold is estimated at a time. Then the maximum expected rounds needed to estimate threshold associated with G_i are $\lfloor \theta_i/\gamma \rfloor (1/\min_{j\in G_i : \mu_j \neq 0} \mu_j)$. Using this fact with definition of A_{θ_n}, the maximum expected rounds needed to estimate n thresholds are $\sum_{i\in A_{\theta_n}} \lfloor \theta_i/\gamma \rfloor (1/\mu_i)$. Once all n thresholds are known then the threshold for any arm k in remaining arms with non-zero mean loss need to iterate only n possible values of thresholds and hence the expected number of rounds needed to its estimate are n/μ_k. Therefore, the maximum number of rounds needed to estimate threshold for all arms in $A_{\theta_n}^c$ are $\sum_{k\in A_{\theta_n}^c : \mu_k \neq 0} n/\mu_k$. With this argument, the proof is complete. □