Improved Regret Bounds with Clustered UCB

Student Paper

No Institute Given

Abstract. In this paper, we present a novel algorithm which achieves a better upper bound on regret for the stochastic multi-armed bandit problem then some of the existing algorithms. Our proposed method clusters arms based on their average estimated payoff. This results in a more efficient elimination of arms within clusters as well as the deletion of weak clusters. We prove the regret upper bound as $\sum_{i \in A} \left(\max \left\{ \left(\frac{27}{c(\Delta_i)^{\frac{3}{5}}} \right), \left(\frac{25\Delta_i}{c\Delta^2)(0.16cT\Delta^2)^{2\Delta/5}} \right) \right\} + \left(\Delta_i + \frac{27 \log (cT \frac{\Delta_i^{\frac{8}{5}}}{12})}{\Delta_i^{\frac{3}{5}}} \right) \right)$, where $c > 0$ is a constant, Δ is the minimal gap between the means of the reward distributions of the optimal arm and a sub-optimal arm, K is the number of arms, T is the horizon and $\psi(m)$ is a parameter of the problem. This bound improves upon the existing algorithm of UCB-Revisited, MOSS, KL-UCB and UCB1 under certain cases. We corroborate our findings both theoretically and empirically and show that by sufficient tuning of parameter, we can achieve a lower regret than all the algorithms mentioned. In particular, in the test-cases where the arm set is dominated by small Δ and large K we achieve a significantly lower cumulative regret than these algorithms.

Keywords: Multi-Armed Bandit, UCB, Exploration-Exploitation, Clustering

1 Introduction

In this paper, we address the stochastic multi-armed bandit problem, a classical problem in sequential decision making. Here, a learning agent is provided with a set of choices or actions, also called arms and it has to choose one arm in each timestep. After choosing an arm the agent receives a reward from the environment, which is an independent draw from a stationary distribution specific to the arm selected. The goal of the agent is to maximize the cumulative reward. The agent is oblivious to the mean of the distributions associated with each arm, denoted by r_i, including the optimal arm which will give it the best average reward, denoted by r^*. The agent records the cumulative reward it has collected for any arm divided by the number of pulls of that arm which is called the estimated mean reward of an arm denoted by $\hat{r}_i$. In each trial the agent faces the exploration-exploitation dilemma, that is, should the agent select the arm which has the highest observed mean reward till now (exploitation), or should the agent explore other arms to gain more knowledge of the true mean reward of the arms and thereby avert a sub-optimal greedy decision (exploration). The objective in the bandit problem is to

maximize cumulative reward which will lead to minimizing the expected cumulative regret. We define the regret(R_T) of an algorithm after T trials as

$$R_T = r^*T - \sum_{i \in A} r_i \mathbb{E}[N_i]$$

where N_i denotes the number of times the learning agent chooses arm i within the first T trials. We also define $\Delta_i = r^* - r_i$, that is the difference between the mean of the optimal arm and the i-th sub-optimal arm (for simplicity we assume that there is only one optimal arm which will give the highest payoff). The problem, gets more difficult when the Δ_i's are smaller and arm set is larger. Also let $\Delta = min_{i \in A} \Delta_i$, that is it is the minimum possible gap over all arms in A.

2 Related Works and Previous Results

Bandit problems have been extensively studied under various conditions. One of the first works can be seen in [18], which deals with choosing between two treatments to administer on patients who come in sequentially. Further studies by [16] and [15], established an asymptotic lower bound for the regret. Lai and Robbins proved in [15] that for any allocation strategy and for any sub-optimal arm i, the regret is lower bounded by

$$\lim_{T \to \infty} \inf \frac{\mathbb{E}[R_T]}{\log T} \geq \sum_{i:r_i < r^*} \frac{(r^* - r_i)}{D(p_i||p^*)}$$

where $D(p_i||p^*)$ is the Kullback-Leibler divergence over the reward density p_i and p^* over the arms having mean r_i and r^*.

Of the several algorithms mentioned in [6], UCB1 has a regret upper bound of $O\left(\frac{K \log T}{\Delta}\right)$ whereas in the same paper the author's propose UCB2 algorithm which has a tighter regret bound than UCB1. UCB2 has a parameter α that needs to be tuned and its regret is upper bounded by $\sum_{i \in A} \left(\frac{(1 + C_1(\alpha)) \log T}{2\Delta_i}\right) + C_2(\alpha)$, where $C_1(\alpha) > 0$ is a constant that can get arbitrarily small when α gets smaller but $C_2(\alpha)$ consequently starts increasing. Another type of strategy was first proposed by [17] called $\epsilon-$greedy strategy where the agent behaves greedily most of the time pulling the arm having highest $\hat{r}_i, \forall i \in A$ and sometimes with a small probability ϵ it will try to explore by pulling a sub-optimal arm. In [6] they further refined the same algorithm and proposed ϵ_n-greedy with regret guarantee. For the ϵ_n-greedy it is proved that if the parameter ϵ, is made a function over time, like $\epsilon_t = \frac{const.K}{d^2 t}$, such that $0 < \epsilon < 1$, then the regret grows logarithmically $\left(\frac{K \log T}{d^2}\right)$. This algorithm performs well given that $0 < d < \min_{i \in A} \Delta_i$ and for large $const$ value the result actually becomes stronger than UCB1. For further insight into various approaches in dealing with stochastic multi-armed bandit we refer the reader to [9]. In [8] they prove an improved regret bounds for the algorithm UCB-Revisited in the order of $\sum_{i \in A} \left(const * \frac{K \log(T\Delta_i^2)}{\Delta_i}\right)$ for very

small Δ_i over a larger set of arms. This is a round based method, where in every round, all the arms are pulled an equal number of times, then based on certain conditions they eliminate some arms and this goes on till one arm is left. In this paper we refer to these type of algorithms as round based algorithms.

Other prominent round based elimination algorithms include Successive Reject, Successive Elimination and Median Elimination. The Successive reject algorithm was proposed by [4], where they explore the pure exploration scenario in a fixed budget/horizon setup. Successive Reject tries to proceed in a phase-wise manner eliminating one arm after each phase. They try to bound the regret by defining two hardness parameter $H_1 = \sum_{i=1}^{K} \frac{1}{\Delta_i^2}$ and $H_2 = \max_{i \in A} i\Delta_i^{-2}$. However, knowledge of these two parameters beforehand is a difficult task so an online approach to estimate them is used in Adaptive UCB-E. The next algorithm called Successive Elimination proposed by [11] also proceeds by eliminating one arm after every round and the authors give PAC-guarantees for them. In PAC guarantee algorithms the learning agent comes up with an ϵ-optimal arm with δ error probability. They also propose a further modification of Successive Elimination called Median Elimination which removes one half of the arms after every round. The sample complexity of Successive Elimination for any sub-optimal arm is bounded by $O\left(\sum_{\Delta_i > \epsilon} \frac{\log(\frac{K}{\delta \Delta_i})}{\Delta_i^2} + \frac{N(\Delta, \epsilon)}{\epsilon^2} \log(\frac{N(\Delta, \epsilon)}{\delta})\right)$, where $N(\Delta, \epsilon)$ are the number of arms which are $\epsilon-$optimal whereas for Median Elimination the sample complexity for any sub-optimal arm is bounded by $O(\sum_{i=1}^{\log_2 K} \frac{K}{\epsilon^2} \log(\frac{1}{\delta}))$, where δ and ϵ are the parameters defined before. For further insight into various approaches in dealing with stochastic multi-armed bandit we refer the reader to [9].

It is also important to distinguish between two approaches in the UCB type algorithms. One approach uses explicit mean estimation using Chernoff Bounds for calculating the confidence bound whereas the other approach uses variance estimation using Bernstein or Bennett's Inequality to estimate the confidence bound. Such variance estimation is found in [6] for the UCB-Normal and UCB-Tuned algorithm and later in [5] where they use Bernstein Inequality to build the confidence term for UCB-V algorithm. But UCB1, UCB2, MOSS, UCB-Revisited, KL-UCB and Median-Elimination uses no such variance estimation techniques. In our analysis also we use no variance estimation methods.

Some of the more recent algorithm like MOSS and KL-UCB provides further refinements to the upper bound in the stochastic multi-armed bandit case. In [3] the MOSS(Minimax Optimal Strategy in Stochastic case) algorithm achieves a distribution free upper bound on the regret as $const.\sqrt{TK}$ and the authors also propose a distribution dependent upper bound as $\sum_{i:\Delta_i>0} \frac{K \log(2 + T\Delta_i^2/K)}{\Delta_i}$. In [13] the authors propose KL-UCB which achieves an upper bound on regret as $\sum_{i:\Delta_i>0} \left(\frac{\Delta_i(1+\alpha)\log T}{D(r_i, r^*)} + C_1 \log\log T + \frac{C_2(\alpha)}{T^{\beta(\alpha)}} \right)$ which is strictly better than UCB1 as we know from Pinsker's inequality $D(r_i, r^*) > 2\Delta_i^2$. KL-UCB beats UCB1, MOSS and UCB-Tuned in various scenarios. Another algorithm that has been proposed in [1] called $UCB(\delta)$ creates

a confidence interval which does not depend on timestep or on horizon T. The regret upper bound in this algorithm is given by $\sum_{i:\Delta_i>0}\left(3\Delta_i + \frac{16}{\Delta_i}\log\left(\frac{2K}{\Delta_i\delta}\right)\right)$, where δ is the error probability as defined before. In [14] the authors come up with the algorithm Deterministic Minimum Empirical Divergence also called DMED+(as referred by [13]) which is first order optimal. This algorithm keeps a track of arms whose empirical mean are close to the optimal arm and takes help of large deviation ideas to find the optimal arm.

Also, we mention the algorithm Exp3 in the adversarial bandit scenario. In the adversarial case, the agent is playing against an adversary who can arbitrarily set a reward on any arm. Algorithms used in adversarial case can be used in the stochastic scenario but not vice-versa. For Exp3 in [7] the authors achieve an upper bound of the order of $O(S\sqrt{KT\log KT})$, where S is defined as the hardness of a problem. Finally, we mention one algorithm that involves Bayesian estimation technique called Thompson Sampling(TS) which uses a prior distribution over arms to estimate the posterior distribution and thereby converge on the optimal arm. In [2] the authors come up with a regret upper bound of the order of $O([K(\frac{1}{\Delta^2})^2]\log T)$ for K-arm stochastic bandits using Thompson Sampling.

3 Contribution and Approach

In this work, we present a novel method where we cluster the arms in each round based on the average estimated payoff $\hat{r}_i$. The purpose of this approach is to control exploration and at the same time exploit the clusters formed to create tight confidence bounds. To do this, we use hierarchical agglomerative clustering using the single-linkage clustering scheme as mentioned in [12]. This is used to group the arms together at the start of each round. Then we deploy a two-pronged approach of exploring inside each clusters to eliminate sub-optimal arms and separately based on other conditions eliminating some weak clusters with all the arms inside it. After each round, at the beginning of the next round we again cluster arms after destroying the old cluster structures formed in the previous round. The logic behind clustering at the beginning of each round afresh is simply that at the initial rounds we have clusters formed with very bad purity level(we can imagine the purity level of a cluster being judged by how many arms having similar or ϵ_m-close means r_i getting clustered together), where in the later rounds we can have tight clusters with high purity level since now we have a better estimate of $\hat{r}_i, \forall i \in A$.

The within cluster arm elimination condition and the entire cluster elimination are two complimentary strategies for speedy elimination of sub-optimal arms. Unlike UCB-Revisited from [8] the within cluster arm elimination leads to dividing the larger problem of finding the optimal arm from the whole arm set into small sub-problems where in each such small clusters (say M clusters, $M \leq K$) we have M arm elimination conditions, thereby increasing the probability of deletion of a sub-optimal arm and hence reducing regret. Also by dividing it into small sub-problems the growth of our pulls in each round is always small given the optimal arm has survived which we ensure by tuning parameters appropriately. We start with a very small cluster size limit(say 2) and double the limit after every round. We put an upper bound on this limit to be decided

by the algorithm online so that the cluster size remains bounded and also since we are exploring each arm in a cluster based on its size, such bounds helps in controlling the exploration within a cluster because we know that single link clustering often forms large chains where the first and large elements may not at all be similar to each other. The entire cluster elimination conditions exploit the idea that if the optimal arm has survived till the later rounds it will be in a cluster of its own with no other sub-optimal arm and then we eliminate all sub-optimal clusters in a single round. This is essentially the same core concept of various round based strategies.

Since, we have assumed that the horizon T is known prior to the agent, hence we introduce the parameter $\psi(m)$ which can be sufficiently tuned to tide over very large or small horizons to enable a balanced exploration by the agent.
Summarizing, the contributions of this research are listed below:

1. We propose a cluster based round-wise algorithm with two arm elimination conditions in each round.
2. We achieved a lower regret bound than UCB-Revisited([8]), UCB(δ) ([1]), UCB1([6]), UCB2([6]), UCB-Tuned([6]), Median Elimination([11]), Exp3([7]) and MOSS ([3] in scenarios when Δ is small and K is large which is encountered frequently in web-advertising domain, which we verify empirically.
3. Our algorithm also compares well with DMED, DMED(+) and KL-UCB.
4. In the critical case when $r_1 = r_2 = .. = r_{K-1} < r^*$ and Δ_is are small, this approach has a significant advantage over other methods.
5. We also come up with an error bound to prove that the error probability decreases exponentially after each round.
6. Unlike KL-UCB our algorithm parameter $\psi(m)$ is not distribution-specific and also our algorithm does not involve calculation of a complex, time consuming function like the divergence function of KL-UCB.

4 Notation Used and Assumptions

In this work, an arm (any arbitrary one) is denoted as a_i and a^* is the optimal arm. The total time horizon is T. Any arbitrary round is denoted by m. The arm set containing all the arms is A, while $|A| = K$. An event is signified by ξ. Any arbitrary cluster is denoted by s_i. The variable S denotes the set of all clusters $s_i \in S$ and $|S| = M \leq K$. The variable ℓ_m is the cluster size limit in each round m and D is the maximum cluster size calculated by the algorithm such that in any round m, $\ell_m \leq D_m$. The variable n_{s_i} denotes the number of times each arm $a_i \in s_i$ is pulled in each round m. The variable B_m denotes the arm set containing the arms that are not eliminated till round m. The variable $\hat{\Delta}_{s,m}$ denotes the empirical gap between the arm having the highest empirical mean and the arm having the lowest empirical mean from B_m in that round m, that is $\hat{\Delta}_{s,m} = \max_{i \in B_m} \{\hat{r}_i\} - \min_{j \in B_m} \{\hat{r}_j\}, i \neq j$ at any round m. In any cluster $s_i \in S$, we denote $\hat{r}_{max_{s_i}} \in s_i$ as the arm with maximum estimated payoff and $\hat{r}_{min_{s_i}} \in s_i$ as the arm with minimum estimated payoff. The parameter $\psi(m)$ is a monotonically decreasing function over the rounds, that is $\psi(m+1) \leq \psi(m)$. We also assume that all rewards are bounded in $[0, 1]$.

The paper is organized as follows, in section 5 we present the algorithm and in section 6 we discuss the algorithm and why it works. Section 7 deals with all the proof including the proofs on regret bound and error bound. In section 8 we present the experimental run of the algorithm and in section 9 we conclude.

5 Clustered UCB

The steps are presented in Algorithm 1.

Algorithm 1: ClusUCB

Pull each arm once and calculate $\hat{r}_i, \forall i \in A$

Let A be such a set which contains all the arms

$B_1 := A$

$\psi(m) = \dfrac{c}{m}, c > 0$

$\ell_1 = 2, D_1 = 2$

For rounds $m = 1, 2, ..\lceil \log T \rceil$

$\hat{\Delta}_{s,m} = \max_{i \in B_m} \{\hat{r}_i\} - \min_{j \in B_m} \{\hat{r}_j\}, i \neq j$

$\epsilon_m = \max\left\{\dfrac{\hat{\Delta}_{s,m}}{\ell_m}, \dfrac{2}{\sqrt{\psi(m)T}}\right\}$, if $\hat{\Delta}_{s,m} \neq 0$

$= \max\left\{\dfrac{1}{D_m}, \dfrac{2}{\sqrt{\psi(m)T}}\right\}$, if $\hat{\Delta}_{s,m} = 0$

Create clusters $s_i = \{a_i\}, \forall i \in B_m$

Cluster s_i, s_j into s_{ij} if $\exists a_i \in s_i$, $a_j \in s_j$, such that $|\hat{r}_i - \hat{r}_j| < \epsilon_m$ and $|s_i| + |s_j| \leq \ell_m$

Pull each arm in s_i,

$n_{s_i} = \left\lceil \dfrac{2\log(\psi(m)T\epsilon_m^2)}{\epsilon_m} \right\rceil$ number of times, $\forall s_i \in S$

Calculate $w_{s_i} = \ell_m^2 k_{s_i}$ $\forall s_i \in S$, where $k_{s_i} = |s_i|$

Arm Elimination:

Delete any arm $a_i \in s_i$ if,

$\hat{r}_i + \sqrt{\dfrac{\log(\psi(m)T\epsilon_m^2)}{2w_{s_i}n_{s_i}}} < \max_{j \in s_i} \hat{r}_j - \sqrt{\dfrac{\log(\psi(m)T\epsilon_m^2)}{2w_{s_i}n_{s_i}}}, \forall s_i \in S$

Cluster Elimination:

Delete any cluster $s_i \in S$ if, $\max_{i \in s_i} \hat{r}_{s_i} + \sqrt{\dfrac{(|B_m|)\epsilon_m \log(\psi(m)T\epsilon_m^2)}{2\ell_m n_{s_i}}}$

$< \min_{j \in s_j : \max \hat{r}_{s_j} \geq \max \hat{r}_{s_i}, \forall s_i \in S} \hat{r}_{s_j} - \sqrt{\dfrac{(|B_m|)\epsilon_m \log(\psi(m)T\epsilon_m^2)}{2\ell_m n_{s_i}}}$

$D_{m+1} = \max\left\{\left\lceil \dfrac{1}{\sqrt{\epsilon_m}} \right\rceil, K\right\}$

$\ell_{m+1} := 2\ell_m$, if $2\ell_m \leq D_{m+1}$

$:= \ell_m$, otherwise

Stopping Condition:

20: Stop the rounds if $|B_m| = 1$ and pull $\max_{i \in B_m} \hat{r}_i$ till T is reached.

6 Discussion on Algorithm and Why it works

In the above algorithm we are dividing the arm set A into smaller clusters which belong to the cluster set S where $|S| \leq K$. We are bounding the cluster size by ℓ_m in each round, starting from an initial value of $\ell_1 = 2$ and doubling it after every round. For creating these clusters we rely on the parameter ϵ_m, which is the range $\hat{\Delta}_{s,m}$(in which all $\hat{r}_i$ lies) divided by ℓ_m. If the range before any round is found to be 0 then we conduct a small exploration with the alternate definition of $\epsilon_m = \frac{1}{D_m}$. Thus we see that as the rounds progresses, ϵ_m gets smaller and smaller resulting in tighter and tighter clusters with high purity level. Single link clustering tends to create large clusters with many elements in a single cluster. To avoid this behavior, mainly because we are conducting exploration inside a cluster based on its size, we bound the maximum cluster size possible in any round by $D_m = \frac{1}{\sqrt{\epsilon_m}}$. Hence as ϵ_m tends to Δ the upper limit on the cluster size gets fixed. On close examination we can see that D_m actually controls our rate of exploration and similar to the ϵ parameter in ϵ-greedy as in [17]. We can also see that at any round m, the pulls n_{s_i} increases at a lesser rate compared to the round-based variants UCB-Revisited $\left\lceil \frac{2log(T\tilde{\Delta}_m^2)}{\tilde{\Delta}_m^2} \right\rceil$, where $\tilde{\Delta}_m$ is initialized at 1 and halved after every round or Median-Elimination $\frac{4}{\epsilon^2} \log\left(\frac{3}{\delta}\right)$, where ϵ, δ are the parameters for PAC guarantee.

Next, in each of the round we eliminate arms like UCB-Revisited([8]) or Median Elimination([11]),however, it is important to note that there is no longer a single point of reference based on which we are eliminating arms but now we have as many reference points to eliminate arms as number of clusters. We can be this aggressive because we have divided the larger problem into smaller sub-problems, doing local exploration and eliminating sub-optimal arms within each clusters with some guarantee. Hence, compared to UCB-Revisited or Median Elimination, the proposed algorithm should have a higher probability of arm deletion. Especially when K is large it is efficient to remove sub-optimal arms quickly rather getting tied down in hopeless exploration. Also our total regret depends on how many arms has survived till m-th round and so we don't need to keep track on the number of clusters formed.

Through cluster elimination condition we ensure that the stopping condition is reached faster. This is a much stricter elimination condition and in the proofs we give a further analysis on why this works. We also define $\psi(m)$ as such a function which monotonically decreases over the rounds that is $|\psi(m+1)| \leq |\psi(m)|$. This is a parameter which we use to tune our exploration and we define its structure later in the proofs. There is also the weight(w_{s_i}) parameter which is calculated online and helps in eliminating arms with a higher probability which we employ to reduce the cumulative regret and this parameter is decide online specific to the cluster $s_i \in S$.

7 Proofs

In this section we will prove the bounds based on the events ξ_1,ξ_2 and ξ_3. In ξ_1, we will assume two important assumptions $i)\hat{r}^* < \hat{r}_i, \forall i \in s_i$ and $ii)\exists a_i \in s_i$ such that $\sqrt{\frac{\epsilon_m}{w_{s_i}}} < \frac{\Delta_i}{5}$. For ξ_2, we will assume that $a^* \in s^*$ and $|s^*| = 1$, $a_i \in s_i \forall a_i \setminus a^* \in B_m$ and $\exists a_{max_{s_i}}$ such that $\sqrt{\epsilon_m} < \frac{2\Delta_s}{5}$, where $\Delta_s = r^* - r_{max_{s_i}}$ and $\hat{r}_{max_{s_i}} > \hat{r}_i, \forall i \in s_i$. ξ_3 be the event when the optimal arm a^* gets eliminated by a sub-optimal arm. At the start of any round m, we fix ϵ_m.

7.1 Theorem 1

Here, we state the main theorem of the paper which shows the regret upper bound of ClusUCB.

Theorem 1. *The upper bound on the total regret over horizon T after round m is given by $R_T \leq \sum_{i \in A} \left(\max \left\{ \left(\frac{27}{c(\Delta_i)^{\frac{3}{5}}} \right), \left(\frac{25\Delta_i}{c\Delta^2)(0.16cT\Delta^2)^{2\Delta/5}} \right) \right\} + \left(\Delta_i + \frac{27 \log (cT \frac{\Delta_i^{\frac{8}{5}}}{12})}{\Delta_i^{\frac{3}{5}}} \right) \right)$, where $c > 0$ is a constant, A is the set of arms and Δ is the minimal gap.*

Remark 1. A sketch of the proof is given here. In the first step, we try to calculate the number of pulls n_{s_i} required to make the optimal arm safe with a high probability so that it becomes the arm with the highest estimated payoff within a cluster. This is shown in Proposition 1. Second step, we try to bound the probability of arm elimination of any sub-optimal arm within a cluster. We try to find out the event which will lead to the elimination of an arm within a cluster. This is shown in Proposition 2 and 3. Third step, we try to bound the probability of cluster elimination with all arms within it and the favourable event leading to it. This is shown in Proposition 4. Finally, in the proof of Theorem 1 we combine all these to get the regret bound.

7.2 Proposition 1

Proposition 1. *The probability that the optimal arm $a^* \in s_i$ will lie above $\hat{r}_{min_{s_i}} + \hat{\Delta}_{s_i}$ after $\left\lceil \frac{2\log(\psi(m)T\epsilon_m^2)}{\epsilon_m} \right\rceil$ pulls in the m-th round is given by $\left\{ 1 - \frac{2}{(\psi(m)T\epsilon_m^2)^{\ell_m^2 \epsilon_m}} \right\}$ where $\hat{r}_{min_{s_i}}$ is the arm with the minimum payoff in s_i, $\hat{\Delta}_{s_i} = max_{i \in s_i} \hat{r}_i - min_{j \in s_i} \hat{r}_j, i \neq j$, $\epsilon_m = \max \left\{ \frac{\hat{\Delta}_{s,m}}{\ell_m}, \frac{1}{\sqrt{\psi(m)T}} \right\}$, if $\hat{\Delta}_{s,m} \neq 0$ and T is the horizon.*

The proof of Proposition 1 is given in **Appendix A**.(Supplementary material)

Remark 2. Thus, we see that as the agent falls through rounds, with increasing values of ℓ_m the probability of optimal arm a^* lying above $\hat{\Delta}_{s_i}$ increases as $\ell_m^2 \epsilon_m \geq \ell_m \Delta$ (as, $\epsilon_m \geq \frac{\Delta}{\ell_m}$) increases with increasing ℓ_m. But, ℓ_m is bounded by D_m and say after the m_D round the probability of optimal arm staying above the specified range gets bounded by $\left\{1 - \frac{2}{(\psi(m)T\epsilon_m^2)^{D_m^2 \epsilon_m}}\right\}$. But we know from the definition of D_m that $D_m = \left\lceil (\frac{1}{\epsilon_m})^{1/2} \right\rceil$ or $D^2 \epsilon_m \approx 1$. Hence, the probability that a^* after n_{s_i} pulls in round m_D going above $\hat{\Delta}_{s_i}$ is $\left\{1 - \frac{2}{(\psi(m)T\epsilon_m^2)}\right\}$.

We must also point out that ϵ_m can become arbitrarily small, so the value $\log(\psi(m)T\epsilon_m^2)$ can become negative. To guard against that scenario we have setup a tolerance level such as $\epsilon_m > \frac{2}{\sqrt{\psi(m)T}}$ and below this value ϵ_m should not be allowed to fall. We also see that ℓ_m is doubled after every round, independent of the rise in D_m and as $\epsilon_m \to \frac{\Delta}{\ell_m}$, ℓ_m gets upper bounded by D_m and will rise no more and hence D_m also gets fixed as $D_m^2 \epsilon_m \approx 1$ for any round m. We make the D_m increase with the rounds as it controls our rate of exploration and in the later rounds we need more exploration as arms close to the optimal arm will survive till the later rounds and the algorithm needs to discriminate amongst them. Also, if ϵ_m is very large, then D_m becomes very small and consequently there is very small exploration. Hence, D_m is the maximum of $\{\frac{1}{\sqrt{\epsilon_m}}, K\}$, which ensures the algorithm does sufficient exploration.

7.3 Proposition 2

Proposition 2. *The number of times an arm $a_i \in s_i$ is pulled in each round is $n_{s_i} = \left\lceil \frac{2\log(\psi m T \epsilon_m^2)}{\epsilon_m} \right\rceil$ and this eliminates the arm a_i such that $\sqrt{\frac{\epsilon_m}{w_{s_i}}} < \frac{\Delta_i}{5}$ by the condition $\left\{\hat{r}_i + \sqrt{\frac{\log(\psi(m)T\epsilon_m^2)}{2w_{s_i}n_{s_i}}} < \max_{j \in s_i} \hat{r}_j - \sqrt{\frac{\log(\psi(m)T\epsilon_m^2)}{2w_{s_i}n_{s_i}}}\right\}, \forall s_i \in S$ with probability $\left\{1 - \left(\frac{1}{2\psi(m)T\epsilon_m^2}\right)\right\}$.*

The proof of Proposition 2 is given in **Appendix B**.(Supplementary material).

Remark 3. Thus, we see that the confidence interval term $c_m = \sqrt{\frac{\log(\psi(m)T\epsilon_m^2)}{2w_{s_i}n_{s_i}}}$ makes the algorithm eliminate an arm a_i as soon as $\sqrt{\frac{\epsilon_m}{w_{s_i}}} < \frac{\Delta_i}{5}$. The above result is in sharp contrast with UCB-Revisited which only deletes an arm if $\tilde{\Delta}_m < \frac{\Delta_i}{2}$, where $\tilde{\Delta}_m$ is initialized at 1 and is halved after every round. We also see from our result that a

much less stricter elimination condition is adopted inside a cluster. We are aggressively eliminating inside a cluster because we are exploring locally and we are guaranteed with $\left\{1-\frac{2}{(\psi(m)T\epsilon_m^2)^{\ell_m^2\epsilon_m}}\right\}$ probability that in the m-th round optimal arm a^* will atleast lie above $\hat{r}_{min_{s_i}}+\hat{\Delta}_{s_i}$. Also, the weight w_{s_i} as shown in the proofs actually help us in faster elimination of arms by the condition ii in ξ_1 explained at the start of the proof. The higher the weight faster is arm elimination but it also increases error probability which might actually lead to a higher cumulative regret. The weight w_{s_i} depends on the cluster size and the larger the cluster size, the higher will be the weight increasing the probability of arm elimination as we want smaller and smaller number of elements in clusters so that we can discriminate effectively amongst the clusters.

7.4 Proposition 3

Proposition 3. *With a probability of* $\left\{1-\left(\frac{2}{\psi(m)T\epsilon_m^2)}\right)\right\}$ *a sub-optimal arm can be deleted within a cluster* s_i *in round* m *by the arm elimination condition, where* $\epsilon_m=\max\left\{\frac{\hat{\Delta}_{s,m}}{\ell_m},\frac{1}{\sqrt{\psi(m)T}}\right\}$*, if* $\hat{\Delta}_{s,m}\neq 0$ *and* T *is the horizon.*

The proof of Proposition 3 is given in **Appendix C**.(Supplementary material)

Remark 4. Thus, we see that the probability of a sub-optimal arm a_i is eliminated in ξ_1 is $\left(1-\frac{2}{\psi(m)T\epsilon_m^2)}\right)$ which increases as the algorithm falls through the round as ϵ_m keeps on decreasing.

Due to the initial uncertainty we can sufficiently tune $\psi(m)$ to increase our pulls in the initial rounds in a bounded fashion. But this function can be set arbitrarily high resulting in a skewed regret upper bound and so we need to define a structure and bound this function as well. We define $\psi(m)=\frac{c}{m}$ where $c>0, m\geq 1$ as defined previously. Notice also that $\psi(m)$ is a monotonically decreasing function and for any round m, $|\psi(m+1)|\leq|\psi(m)|$. When the time horizon T is very large or very small, as the pulls n_{s_i} depends on T, we can tune $\psi(m)$ by changing c so that the exploration remains balanced. Since we are exploring locally and the pulls are increasing after every round so we decrease $\psi(m)$ in the later rounds which not only tapers down the growth of n_{s_i} but also decreases arm elimination probability, as in the later rounds, only arms closer to the optimal arm survive and we need careful elimination.

7.5 Proposition 4

Proposition 4. *With a probability of* $\left(1-\frac{4}{(\psi(m)T\epsilon_m^2)^{1+|B_m|^2\epsilon_m}}\right)$ *a sub-optimal arm can be deleted in round* m*, where* $\hat{\Delta}_{s,m}=\max_{i\in B_m}\{\hat{r}_i\}-\min_{j\in B_m}\{\hat{r}_j\}$ *for* $i\neq j$*,* $\epsilon_m=\max\left\{\frac{\hat{\Delta}_{s,m}}{\ell_m},\frac{1}{\sqrt{\psi(m)T}}\right\}$ *for* $\hat{\Delta}_{s,m}\neq 0$*,* B_m *is the set of arms still not eliminated in the* m*-th round and* T *is the horizon.*

The proof of Proposition 4 is given in **Appendix D**.(Supplementary material)

Remark 5. Thus, from the conditions imposed on ξ_1 and ξ_2 we see that the cluster deletion condition is more tightly coupled than arm elimination condition. This is because, for cluster elimination the condition depends on the number of arms surviving till m-th round that is $|B_m|$ which itself depends on the pulls n_{s_i} and cluster size limit D_m. We also point out that at any round m there will be $|S_m|$ arm elimination conditions because each cluster has its own arm elimination condition weighted by w_{s_i}, resulting in the probability of $\max\left\{\left(\frac{2}{\psi(m)T\epsilon_m^2}\right),\left(\frac{4}{(\psi(m)T\epsilon_m^2)^{1+|B_m|^2\epsilon_m}}\right)\right\}$ of being eliminated, an increase over UCB-Revisited and Median Elimination. This is because each arm can either be eliminated by the the arm elimination condition within a cluster or by the cluster elimination condition whereby all the arms within a cluster are eliminated. Also, we see that lower bounding $\epsilon_m = \frac{2}{\sqrt{\psi(m)T}}$, lower bounds the probability of arm elimination atleast 0.5 in ξ_1 and in ξ_2 the probability of cluster elimination and stopping very close to 1.

7.6 Proof of Theorem 1

The proof of Theorem 1 is given in **Appendix E**.(Supplementary material)

Remark 6. Thus, we see the most significant term in the regret is $\frac{27\log(cT\frac{\Delta_i^{\frac{8}{5}}}{12})}{\Delta_i^{\frac{3}{5}}}$, which is significantly lower than UCB1, UCB2, EXP3, UCB(δ), MOSS, UCB-Revisited and Median Elimination under certain cases when the $\Delta \to 0$ and K is large. Also ClusUCB is more efficient than UCB1, MOSS, EXP3, UCB(δ), KL-UCB as K scales up because being a round-based algorithm, it is removing sub-optimal arms in each round as opposed to the former algorithms which are calculating the confidence bounds over all arms in each timestep and then choosing the max of them. Also evident from the proof is that if the horizon T is small or very large we can tune the parameter $\psi(m)$ carefully to have a lesser regret.

7.7 Error Bound

In this section, we try to come up with an error bound for the algorithm if at any round m, the optimal arm a^* gets eliminated by another sub-optimal arm. Since, arm elimination condition is the more aggressive elimination condition, we come up with an error bound that bounds the regret once the optimal arm gets eliminated by another sub-optimal arm. The proof of this directly follows from **Theorem 2**, and mimics the proof as in [8].

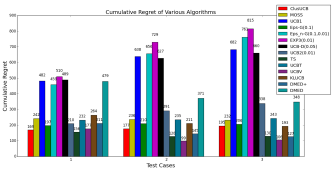

Fig. 1. Experiment 1: Regret for various Algorithms in 3 testcases. $T = 20000; \psi(m) = 1.5/m$

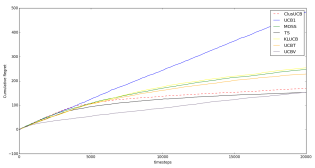

Fig. 2. Experiment 2: Growth of Regret for test case 1. $T = 20000; \psi(m) = 1.5/m$

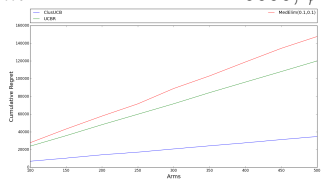

Fig. 3. Experiment 3: Regret for ClusUCB, UCB-Revisited and Median-Elimination. $T = 5 \times 10^5; \psi(m) = 0.1/m$

Theorem 2. *The error bound till round* m *is given by* $e_t \leq \sum_{i \in A'} \left(\frac{51}{\psi(m)\Delta_i^{6/5}} \right) + \sum_{i \in A'' \setminus A'} \left(\frac{51}{\psi(m)\Delta_b^{6/5}} \right)$*, where the arms surviving till* m*-th round belong to the set* $A^{'}$*, arms to still survive and eliminate arm* a^* *after round* m *belong to* $A^{''}$*.*

The proof of theorem 5 is given in **Appendix F**.(Supplementary material)

8 Experimental Run

The first experiment is conducted over a testbed of 10 arms for the 3 test-cases involving Bernoulli reward distribution with expected rewards of the arms as described below.

Test Case 1: [0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.07, 0.1]
Test Case 2: [0.07, 0.07, 0.07, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.1]
Test Case 3: [0.07, 0.07, 0.07, 0.05, 0.05, 0.05, 0.03, 0.03, 0.03, 0.1]

These type of cases are frequently encountered in web-advertising domain. The regret is averaged over 50 independent runs over each testbed and is shown in **Fig 1**. 14 algorithms ClusUCB, MOSS, UCB1, UCB2($\alpha = 0.01$), ϵ-greedy($\epsilon = 0.1$), ϵ_n-greedy($c = 0.1, d = 0.01$), Exp3($\gamma = 0.01$), UCB-Delta($\delta = 0.05$), UCB-Tuned, UCB-V, KL-UCB, DMED+, DMED(as stated in [13]) and Thompson Sampling are run over this testbed and shown in this figure(the cumulative regret averaged over 50 independent runs is shown above each bar). Here, we see that except Thompson Sampling and UCB-V and DMED+ the regret of Clustered-UCB is lower than the rest for

all the test cases. Even in test case 1, the regret of ClusUCB is nearly same as Thompson Sampling and better than UCB-V and DMED+. In test case 1 the regret is so low for ClusUCB because $\Delta_i = \Delta, \forall i \in A$ and while all the algorithms employ significant exploration ClusUCB by virtue of dividing the problem into sub-problems quickly finds the optimal arm. The parameters of ϵ_n-greedy are very difficult to estimate and if d is not a tight lower bound of Δ then the result can be poor as shown in the figure. The parameter less algorithms UCB1, MOSS, UCB-Tuned, Thompson Sampling and UCB-V are run as mentioned in the respective papers. For algorithms requiring parameters, such as UCB2, UCB-Delta, ϵ-greedy, ϵ_n-greedy, EXP3, the parameters were tried and tested over several values and then implemented in each of the test cases. KL-UCB, DMED and DMED+ code is taken from [10], and is run accordingly the way author specified with KL-UCB parameter $c = 0$. KL-UCB regret is also poorer than ClusUCB in test case 1 and 2 and same as ClusUCB in test case 3. It takes significant more time to run the algorithm than ClusUCB which might not be feasible in many real world scenarios like web advertising. In this short horizon for all the test cases ClusUCB performs better than MOSS and UCB1. DMED+, UCB-V and TS, as expected beats ClusUCB in scenario 2 and 3. In test cases 2 and 3, ClusUCB performs bad mainly because here the Δ_i is much more evenly spread and ClusUCB takes significantly more exploration to find the optimal arm. Also since it's a short horizon of $T = 20000$, we have taken $\psi(m) = \dfrac{1.5}{m}$.

The second experiment is conducted over the same testbed as in test case 1 above and shown in **Fig 2**. We check the growth of regret over time for 7 algorithms as mentioned in the figure. Here, we see that ClusUCB has a much steeper regret curve than the rest which signifies a faster exploitation and less exploration. It quickly finds the optimal arm and the cumulative regret nearly becomes negligible. UCB1 is not able to find the optimal arm in this short horizon whereas UCB-V performs much better but its regret still does not stabilize within this short horizon. TS performs well as well whereas MOSS performs much worse and hardly stabilizes in this short horizon. We also see that ClusUCB is remarkably stable in this short horizon and it outputed a sub-optimal arm only 5 out of 50 runs. Comparing this with UCB-V, it outputed a wrong arm 11 out of 50 runs.

The third experiment is conducted over a testbed of $100 - 500$ arms at an interval of 50 arms, of which (for any particular run) $\frac{1}{3}$ arms have Gaussian reward distribution $N(\mu = 0.2, \sigma = 0.3)$, rest $\frac{2}{3}$ arms have Gaussian reward distribution $N(\mu = 0.7, \sigma = 0.3)$ and the optimal arm have parameters $N(\mu = 0.9, \sigma = 0.3)$. We conduct this experiment to not only check the performance of ClusUCB in Gaussian distribution, but also the growth of regret over a large set of arms. Here, we employ only select algorithms(round based algorithms) since the action space is very large and the other algorithms will take a long time to converge on the optimal arm. The regret is averaged over 50 independent runs over this testbed and is shown in **Fig 3**. Three round-based algorithms ClusUCB, UCB-Revisited and Median Elimination are run over this testbed and shown in the figure. Here, Median-Elimination performs worse than UCB-Revisited. We also see that over this large set of arms the regret of ClusUCB is not only much lesser than UCB-Revisited and Median-Elimination but also being a round

based algorithm it is much faster than other standard algorithms. Again since it's a large horizon of $T = 5 \times 10^5$ so we have taken $\psi(m) = \dfrac{0.1}{m}$.

9 Conclusion and Further Discussion

Our study concludes that the regret of ClusUCB is lower than UCB1, UCB2, EXP3, MOSS, UCB-Revisited, KL-UCB, UCB-Tuned, DMED and Median Elimination under certain cases when the $\Delta \to 0$ and K is large. Such cases can frequently occur in web advertising scenarios where the r_i are small and a large number of ads are available in the pool. We see that the upper bound of ClusUCB has $\log$ dependence on horizon T whereas MOSS has a dependence of $T^{1/2}$ in the distribution free case. For the critical case when $\Delta_{i:r_i<r*}, \forall i \in A$ are equal then ClusUCB performs better than UCB-Revisited, UCB1, UCB-Tuned, KL-UCB, MOSS and EXP3 with sufficient tuning of parameters as shown empirically. Also ClusUCB scales well with large K as compared with UCB1, EXP3, MOSS, KL-UCB and Thompson Sampling since it is eliminating sub-optimal arms after some timesteps. We must also remember as the number of arms increases UCB1, UCB2, KL-UCB and UCB(δ) will take more time as all of these algorithms have to build their confidence set over all the arms which will take $O(K)$ time whereas algorithms like ClusUCB, UCB-Revisited and Median Elimination will take much lesser time as they keep on removing sub-optimal arm after each round. KL-UCB because of its calculation of the divergence function performs much slower as compared to ClusUCB. Also, ClusUCB can be used in the budgeted bandit setup since within a fixed horizon/budget T, the algorithm comes up with an optimal arm with an exponentially decreasing error probability as proved theoretically in this work. Further uses of this algorithm can be in the contextual bandit scenario whereby the clustering of arms can be done on the basis of the feature vectors of the arms and users.

References

1. Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.
2. Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. *arXiv preprint arXiv:1111.1797*, 2011.
3. Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.
4. Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on Learning Theory-2010*, pages 13–p, 2010.
5. Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

6. Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.
7. Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
8. Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.
9. Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.
10. Olivier Cappe, Aurelien Garivier, and Emilie Kaufmann. pymabandits, 2012. http://mloss.org/software/view/415/.
11. Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105, 2006.
12. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.
13. Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. *arXiv preprint arXiv:1102.2490*, 2011.
14. Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *COLT*, pages 67–79. Citeseer, 2010.
15. Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.
16. Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1952.
17. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.
18. William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.