Anytime Exploration for Multi-armed Bandits using Confidence Information

Kwang-Sung Jun KJUN@DISCOVERY.WISC.EDU
Wisconsin Institutes for Discovery, UW-Madison, 330 N. Orchard St., Madison, WI 53715 USA

Robert Nowak RDNOWAK@WISC.EDU
Wisconsin Institutes for Discovery, UW-Madison, 330 N. Orchard St., Madison, WI 53715 USA

Abstract

We introduce anytime Explore-m, a pure exploration problem for multi-armed bandits (MAB) that requires making a prediction of the top-m arms at every time step. Anytime Explore-m is more practical than fixed budget or fixed confidence formulations of the top-m problem, since many applications involve a finite, but unpredictable, budget. However, the development and analysis of anytime algorithms present many challenges. We propose AT-LUCB (AnyTime Lower and Upper Confidence Bound), the first nontrivial algorithm that provably solves anytime Explore-m. Our analysis shows that the sample complexity of AT-LUCB is competitive to anytime variants of existing algorithms. Moreover, our empirical evaluation on AT-LUCB shows that AT-LUCB performs as well as or better than state-of-the-art baseline methods for anytime Explore-m.

1. Introduction

We consider the top-m arms identification problem for multi-armed bandits (MAB). Suppose we have n stochastic arms. When an arm i is pulled, a reward is drawn i.i.d. from a distribution ν_i whose support is in $[0, 1]$.[1] We define the expected reward of arm i as $\mu_i := \mathbb{E}_{X\sim\nu_i} X$. Without loss of generality, we assume $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m > \mu_{m+1} \geq \cdots \geq \mu_n$. The goal is to find the set of m arms with the highest expected rewards $(\mu_1, \ldots, \mu_m)$ through efficient sampling decisions.

The top-m arms identification problem for MABs has a long history dating back to the '50s (Bechhofer, 1958; Paulson, 1964). The vast majority of the recent studies fall under either the fixed confidence or the fixed budget setting. The fixed confidence setting asks an algorithm to take a failure rate δ and recommend the top-m arms with probability at least $1 - \delta$ using as few samples as possible (Even-dar et al., 2002; Kalyanakrishnan et al., 2012; Karnin et al., 2013; Jamieson et al., 2014). The fixed budget setting asks an algorithm to sample B times only and then recommend the top-m with the highest possible confidence (Audibert et al., 2010; Karnin et al., 2013; Bubeck et al., 2013). In both settings, the user provides an algorithm with δ or B based on which the algorithm adjusts the level of aggressiveness in its adaptive sampling strategy.

Deviating from the two settings above, we introduce a new setting where an algorithm must recommend the top-m arms after every time step, which we call **anytime Explore-*m***. Anytime Explore-*m* generalizes the anytime setting of (Bubeck et al., 2009) from identifying the best arm to the top-m arms. Specifically, an algorithm for anytime Explore-*m* must perform at each time step t the following two tasks. First, the algorithm must choose which arm to pull (I_t) based on the samples collected so far. After receiving a reward from arm I_t, second, the algorithm must choose and output a set $J(t)$ of m arms that are believed to be the top-m. The user does not provide a desired failure rate δ nor a budget B; an algorithm must adapt its aggressiveness of the sampling over time. Note that anytime Explore-*m* asks a strictly harder question than the fixed budget setting in that an anytime algorithm can perform under the fixed budget setting but not vice versa.

Anytime Explore-*m* is more challenging than the fixed confidence and budget settings, but it is a better fit for many practical problems. For example, The New Yorker cartoon caption contest[2] uses crowdsourcing to collect ratings for the hundreds or thousands of captions submitted for each week's cartoon. The ratings are used to select the top-m captions (e.g., $m = 50$) to be advanced to the final round of in-depth evaluation. The crowdsourcing system is based on NEXT (Jamieson et al., 2015), and it adaptively selects captions for ratings using a fixed-confidence algo-

[1] This can be generalized to a sub-Gaussian distribution while keeping the means in $[0, 1]$.

[2] http://contest.newyorker.com

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

rithm (Jamieson et al., 2014) in order to home-in on the top captions as quickly as possible. In this application, however, the total number of ratings that will be collected is both *limited* and *unknown* ahead of time (specifically, the crowdsourcing is carried out over a fixed period of time, but the number of responses collected in that time is unpredictable). Since the number of ratings is limited, fixed-confidence algorithms are not ideal, and since the number of ratings is unknown, fixed-budget algorithms are not appropriate. The New Yorker crowdsourcing task is precisely the anytime Explore-m problem.

Let $J^\star := \{1, 2, \ldots, m\}$, the true top-$m$ set. We are interested in bounding the misidentification probability

$$\mathbb{P}(J(t) \neq J^\star)$$

for every $t \geq 1$ and we would like the bound to decay with t. (Bubeck et al., 2009) introduced anytime Explore-1 and analyzed the expected simple regret $\mathbb{E}[\mu_1 - \mu_{J(t)}]$ instead of the misidentification probability. For $m = 1$, the two quantities are closely related, and the misidentification probability can be easily derived from the analyses therein; see our supplementary material.

There exists a trivial way to construct an anytime algorithm with a fixed-budget algorithm such as Successive Accepts and Rejects (SAR) (Bubeck et al., 2013); apply the so-called "doubling trick" on the budget. That is, for stage $s = 1, 2, \ldots$, run SAR with budget $2^{s-1}n$ while recommending at each time step the empirical top-m arms from the end of the previous stage rather than from the current time step. One can easily show that this trick yields an anytime algorithm with essentially the same performance guarantees at time $t = B$ as standard SAR with budget B; see our supplementary material. However, this trick is a rather trivial extension and does not exploit confidence bound information, which we argue can be quite helpful. Fixed-confidence algorithms also exist in top-m identification, but we are not aware of confidence-based anytime algorithms.[3]

We propose a new algorithm AT-LUCB that solves anytime Explore-m. AT-LUCB is a variant of the LUCB algorithm (Kalyanakrishnan et al., 2012) that adapts the failure rate parameter δ over time in a data-dependent manner. Put another way, we repeatedly run LUCB while geometrically reducing δ after each run. The analysis is not as trivial as doubling the budget of the fixed-budget algorithms since the length of each LUCB run is stochastic. We elaborate why the analysis is not trivial in Section 2 as well as present and analyze AT-LUCB. Furthermore, our experiments in Section 3 show that AT-LUCB performs as well as or better than algorithms based on the budget doubling trick.

[3] UCB-E (Audibert et al., 2010) and KL-LUCB-E (Kaufmann & Kalyanakrishnan, 2013) that works for the fixed budget setting use the confidence bounds but requires the problem hardness parameter as an input, which is unrealistic. UCB (Bubeck et al., 2009) is for $m = 1$ only.

Algorithm	Sample complexity
Uniform	$O\left(n\left(\Delta_m^{\langle m\rangle}\right)^{-2}\ln\left(\frac{n}{\delta}\right)\right)$
Doubling SAR	$O\left(H_2^{\langle m\rangle}\ln(n)\ln\left(\frac{n}{\delta}\right)\right)$
AT-LUCB	$O\left(H^{\langle m\rangle}\max\left\{\ln\left(\frac{H^{\langle m\rangle}}{\delta}\right), \frac{\ln^2\left(\frac{1}{\delta}\right)}{\ln(H^{\langle m\rangle})}\right\}\right)$

Table 1. Comparison of anytime Explore-m algorithms

Define the gap parameters $\Delta_i^{\langle m\rangle} := \mu_i - \mu_{m+1}, \forall i \leq m$ and $\Delta_i^{\langle m\rangle} := \mu_m - \mu_i, \forall i \geq m+1$. Let σ be a permutation that sorts the arms in increasing order of the gaps: $\Delta_{\sigma(1)}^{\langle m\rangle} = \Delta_{\sigma(2)}^{\langle m\rangle} \leq \Delta_{\sigma(3)}^{\langle m\rangle} \leq \ldots \leq \Delta_{\sigma(n)}^{\langle m\rangle}$. We define the problem hardness parameters $H^{\langle m\rangle} := \sum_{i=1}^n (\Delta_i^{\langle m\rangle})^{-2}$ and $H_2^{\langle m\rangle} := \max_{i\in\{1,\ldots,n\}} i\left(\Delta_{\sigma(i)}^{\langle m\rangle}\right)^{-2}$. $H^{\langle m\rangle}$ and $H_2^{\langle m\rangle}$ are closely related (explained below). *Sample complexity* is the smallest number T of samples required to achieve the target level δ of misidentification probability: $\mathbb{P}(J(T) \neq J^\star) \leq \delta$. For comparing algorithms, it is more straightforward to discuss the sample complexity rather than the misidentification probability. Table 1 compares the sample complexities of the various anytime algorithms.

Uniform is a trivial algorithm that samples the least pulled arm at each time step. The bounds in Table 1 show that AT-LUCB is better than Uniform since $n(\Delta_m^{\langle m\rangle})^{-2} = n\max_i((\Delta_i^{\langle m\rangle})^{-2})$ is much larger than $H^{\langle m\rangle}$ in general. We claim that the sample complexity of AT-LUCB is better than or equal to that of doubling SAR. In Section 2, we show that for even small problems like $n = 10$ the first term in the max of AT-LUCB's complexity dominates the second. Thus, the sample complexity of AT-LUCB is practically $O(H^{\langle m\rangle}\ln(H^{\langle m\rangle}/\delta))$. To illustrate the difference between the complexities of AT-LUCB and Doubling SAR, suppose, for example, that the gaps follow $\Delta_{\sigma(i)}^{\langle m\rangle} \propto (i/n)^\beta, \forall i \geq 2$ for some $\beta > 0$. If $\beta < 1/2$, then $H^{\langle m\rangle} = O(n)$. If $\beta = 1/2$, then $H^{\langle m\rangle} = O(n\ln n)$. If $\beta > 1/2$, then $H^{\langle m\rangle} = O(n^{2\beta})$. In all these cases, $\ln(H^{\langle m\rangle}) = O(\ln(n))$, which is better than $O(\ln^2(n))$ that appear in doubling SAR. One can also show that $H^{\langle m\rangle}/\ln(2n) \leq H_2^{\langle m\rangle} \leq H^{\langle m\rangle}$. If $H_2^{\langle m\rangle} \approx H^{\langle m\rangle}/\ln(2n)$, then the sample complexities of AT-LUCB and SAR are of the same order. However, if $H_2^{\langle m\rangle} \approx H^{\langle m\rangle}$, then AT-LUCB is $\ln(n)$ factor better.

For the special case of $m = 1$, we mention a few related algorithms. The UCB algorithm of (Bubeck et al., 2009) with exploration parameter $\alpha \leftarrow 2$ has the sample complexity $O\left(\max\left\{n(\Delta_1^{\langle 1\rangle})^{-2}\ln(n(\Delta_1^{\langle 1\rangle})^{-2}), n^{\frac{3}{2}}(1/\delta)^{\frac{1}{2}}, n^2\right\}\right)$. This is suboptimal due to the polynomial dependence on

$1/\delta$ (rather than logarithmic) and the term $n(\Delta_1^{\langle 1\rangle})^{-2}$ rather than $H^{\langle m\rangle}$. Also, one can apply the same doubling trick to Successive Halving (SH) (Karnin et al., 2013). Doubling SH has the sample complexity $O\big(H_2^{\langle 1\rangle}\ln(n)\ln\big(\frac{\ln n}{\delta}\big)\big)$, which is slightly better than doubling SAR with $m=1$. Under the polynomial decay model for the gaps considered above, the sample complexity of AT-LUCB with $m=1$ can be slightly inferior to doubling SH (by a factor of at most $\ln(n)/\ln\ln(n)$) or can be $\ln(\ln(n))$ factor better, depending on $H_2^{\langle m\rangle}$. The details on deriving the sample complexities of Uniform, UCB, and doubling SH are found in our supplementary material.

2. AT-LUCB Algorithm

We propose AT-LUCB and prove its anytime guarantee. AT-LUCB uses the LUCB algorithm (Kalyanakrishnan et al., 2012) as a subroutine. We first review LUCB. LUCB is a fixed-confidence algorithm that, given a failure rate δ, performs adaptive sampling in order to identify the top-m arms with probability at least $1-\delta$. LUCB uses a deviation function

$$\beta(u,t,\delta) := \sqrt{\frac{1}{2u}\ln\left(\frac{k_1 n t^4}{\delta}\right)}, \text{ where } k_1 = \frac{5}{4}.$$

Alternatively, one can use a tighter bound described in (Kaufmann & Kalyanakrishnan, 2013) for a better performance. We define $\beta(0,t,\delta) := \infty$ for convenience. We also define the following.

Key Quantities

$\widehat{\mu}_a^t$ and u_a^t, the empirical mean and number of samples of arm a, respectively, at the end of time $t-1$

$L_a^t(\delta) := \widehat{\mu}_a^t - \beta(u_a^t,t,\delta)$, lower confidence bound

$U_a^t(\delta) := \widehat{\mu}_a^t + \beta(u_a^t,t,\delta)$, upper confidence bound

High^t, empirical top-m arms at the end of time $t-1$

$h_*^t(\delta) := \arg\min_{a\in\text{High}^t} L_a^t(\delta)$

$\ell_*^t(\delta) := \arg\max_{a\notin\text{High}^t} U_a^t(\delta)$

At each time step t, the algorithm pulls both $h_*^t(\delta)$ and $\ell_*^t(\delta)$. This means that the total number of samples are twice the number of time steps proceeded. Define the terminating condition

$$\text{Term}^t(\delta,\epsilon) := \{U^t_{\ell_*^t(\delta)}(\delta) - L^t_{h_*^t(\delta)}(\delta) < \epsilon\}, \quad (1)$$

where $\epsilon \geq 0$. Let $J_\epsilon^\star := \{a \in \text{Arms} \mid \mu_a \geq \mu_m - \epsilon\}$. When $\text{Term}^t(\delta,\epsilon)$ is satisfied, LUCB terminates and outputs the empirical top-m arms whose means are guaranteed to be at least $\mu_m - \epsilon$ with probability at least $1-\delta$: $\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star \mid \text{Term}^t(\delta,\epsilon)) \leq \delta$. We refer to (Kalyanakrishnan et al., 2012) for details on LUCB.

We focus on the case $\epsilon = 0$. Then, $J^\star = J_0^\star$ and so $\mathbb{P}(J(t) \not\subseteq J_0^\star) = \mathbb{P}(J(t) \neq J^\star)$. However, we carry the symbol ϵ for generality.

Algorithm 1 AT-LUCB

Input: n arms, m: the target number of top arms, $\delta_1 \leq [1/200, n]$, $\alpha \in [1/50, 1)$, $\epsilon \geq 0$
Output: m arms.
$t \leftarrow 1$, $S(0) \leftarrow 1$, $\delta_s \leftarrow \delta_1 \alpha^{s-1}, \forall s \geq 1$
while True **do**
  if $\text{Term}^t(\delta_{S(t-1)}, \epsilon)$ **then**
    $S(t) \leftarrow \max\{s' \geq S(t-1)+1 : \neg\text{Term}^t(\delta_{s'}, \epsilon)\}$
    $J(t) \leftarrow \{\text{the empirical top-}m \text{ arms}\}$
  else
    $S(t) \leftarrow S(t-1)$
    $J(t) \leftarrow J(t-1)$ (or the empirical top-m arms if $S(t)=1$)
  end if
  Pull $h_*^t(\delta_{S(t)})$ and $\ell_*^t(\delta_{S(t)})$
  Recommend $J(t)$.
  $t \leftarrow t+1$
end while

The main idea of AT-LUCB is to repeatedly run LUCB while geometrically reducing the failure rate parameter δ after each run. Let

$$\delta_s := \delta_1 \alpha^{s-1}, \quad (2)$$

where $s \geq 1$ is the stage index, $\delta_1 \in [1/200, n]$ is the initial failure rate and $\alpha \in [1/50, 1)$ is the discount factor. For stage $s = 1, 2, \ldots$, we run LUCB with failure rate δ_s until satisfying the stopping criterion $\text{Term}^t(\delta_s, \epsilon)$. Note that the length of the stage is not deterministic but stochastic. In any stage, empirical means are computed based the samples collected so far including all the previous stages. Recall that an anytime Explore-m algorithm must predict the top-m arms at every time t. Let $S(t)$ be the stage to which time step t belongs. At every time step t, AT-LUCB recommends the empirical top-m arms computed at the end of the previous stage $S(t)-1$. If $S(t)=1$, then AT-LUCB recommends the current empirical top-m arms. We present the pseudocode of AT-LUCB in Algorithm 1.

A simple, but naive, bound on $\mathbb{P}(J(t) \neq J^\star)$ could be obtained from Corollary 7 in (Kalyanakrishnan et al., 2012), as follows. That corollary shows that when t is sufficiently large, the probability of LUCB not terminating after t time steps is upperbounded by $4\delta/t^2$. Consider

$$\mathbb{P}(J(t) \neq J^\star) = \sum_{s\geq 1} \mathbb{P}(J(t) \neq J^\star \mid S(t) = s)\mathbb{P}(S(t) = s),$$

and note that the bound above only guarantees that $P(S(t) \leq 1) \leq 4\delta_1/t^2$. Clearly, this is insufficient, since the bound only decays polynomially in t, rather than exponentially as in doubling SAR.

Main results Let $x \vee y := \max\{x, y\}$. We define ϵ-tolerant problem hardness $H^{\epsilon/2} := \sum_{a=1}^{n} \left(\Delta_a^{\langle m \rangle} \vee \frac{\epsilon}{2}\right)^{-2}$, which is equivalent to $H^{\langle m \rangle}$ when $\epsilon = 0$. Define

$$\gamma_s^* := \max\left\{\frac{3}{4}, 1 - \frac{\ln\left(\frac{12k_1 n H^{\epsilon/2}}{\delta_1}\right)}{\ln\left(\frac{12k_1 n H^{\epsilon/2}}{\delta_s}\right)}\right\} \quad \text{and} \tag{3}$$

$$T_s^* := \left\lceil \frac{24H^{\epsilon/2}}{1-\gamma_s^*} \ln\left(\frac{12k_1}{1-\gamma_s^*}\frac{nH^{\epsilon/2}}{\delta_s}\right)\right\rceil . \tag{4}$$

Here is the intuition for our analysis of AT-LUCB. Let $T_s^0 := \lceil 146H^{\epsilon/2}\ln(H^{\epsilon/2}/\delta_s)\rceil$. We know from the theory of the standard LUCB with parameter δ_s that with probability at least $1-\delta_s$ it will terminate within T_s^0 time steps and be correct with the same probability. With this in mind, after T_s^0 time steps we wish to show that AT-LUCB has finished the stage s with probability at least $1-\delta_s$. That is, at that time step, AT-LUCB is performing about as well as the δ_s-confidence LUCB algorithm. Showing that all earlier stages ($\le s$) of AT-LUCB have terminated with probability at least $1-\delta_s$ is quite delicate and is the main contribution of our analysis. In doing so, we replace T_s^0 with T_s^*, which is slightly larger than T_s^0 due to a technical reason.

Theorem 1 states that T_s^* is a sufficient number of time steps that guarantees the misidentification probability to be under $2\delta_s$. Define $[a..b] := \{a, a+1, \ldots, b\}$.

Theorem 1. *In AT-LUCB,* $\forall s \ge 1, \forall t \in [T_s^*..(T_{s+1}^*-1)]$,

$$\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star) \le 2\delta_s .$$

Proof. We present the sketch of the proof here; refer to Section 4 for detail. Let $t \in [T_s^*..(T_{s+1}^*-1)]$ for some $s \ge 1$. Note that

$$\begin{aligned}
&\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star) \\
&\quad = \mathbb{P}(S(t) \ge s+1)\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star \mid S(t) \ge s+1) + \\
&\qquad \mathbb{P}(S(t) \le s)\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star \mid S(t) \le s) . \\
&\quad \le 1 \cdot \mathbb{P}(J(t) \not\subseteq J_\epsilon^\star \mid S(t) \ge s+1) + \mathbb{P}(S(t) \le s) \cdot 1.
\end{aligned}$$

We claim that $\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star \mid S(t) \ge s+1) \le \delta_s$. Suppose $S(t) = s+1$ for simplicity. When the previous stage s ended (say at time t'), the output $J(t')$ (the same as $J(t)$) is not the true top-m arms with probability at most δ_s due to the stage-terminating condition $\mathrm{Term}^t(\delta_s, \epsilon)$.

Then, it remains to show that $\forall s \ge 1$,

$$t \in [T_s^*..(T_{s+1}^*-1)] \implies \mathbb{P}(S(t) \le s) \le \delta_s . \tag{5}$$

One can show (5) for $s = 1$ using Lemma 5 of (Kalyanakrishnan et al., 2012). However, proving (5) for $s \ge 2$ is nontrivial. Let us consider $s = 2$ first. Lemma 5 therein introduces a "bad" event with the parameter δ (fixed) and bounds its probability. We replace the δ by $\delta_{S(t)}$ which is now a *random variable* and show (5) for $s = 2$ where we need the fact that (5) is true with $s = 1$. We generalize this idea and apply induction to prove (5) for every $s \ge 1$. □

Note that Theorem 1 bounds the misidentification probability by a piecewise constant function. To make it easier to comprehend, Corollary 1 finds a strictly decreasing (in t) upperbound of the piecewise constant function.

Corollary 1. *In AT-LUCB, for all* $t \ge T_1^*$,

$$\mathbb{P}(J(t) \not\subseteq J_\epsilon^\star) \le \max\left\{\frac{96k_1 n H^{\epsilon/2}}{\alpha}\exp\left(-\frac{t-1}{96H^{\epsilon/2}}\right), \frac{24k_1 n H^{\epsilon/2}}{\alpha}\exp\left(-\sqrt{\frac{(t-1)\ln(12k_1 n H^{\epsilon/2}/\delta_1)}{36H^{\epsilon/2}}}\right)\right\} . \tag{6}$$

Corollary 1 shows that the misidentification probability bound behaves like $\exp(-t)$ for smaller t and $\exp(-\sqrt{t})$ for larger t. We claim that such a transition of the bound happens only after the bound becomes too small to care. To see this, verify that $t \le 1 + 256H^{\epsilon/2}\ln(12k_1 n H^{\epsilon/2}/\delta_1) =: t'$ implies that the max in (6) is achieved with the first element. When $t = t'$, the maximum is $\frac{8\delta_1^{8/3}}{\alpha(12k_1 n H^{\epsilon/2})^{5/3}}$. Even for a small problem like $n = 10$, the maximum is extremely small. For example, plugging in $H^{\epsilon/2} \leftarrow n$, $\alpha \leftarrow 1/2$, and $\delta_1 \leftarrow n$ implies that the bound is less than 10^{-10}. In other words, the second element of the max in (6) dominates only after the bound becomes smaller than 10^{-10}.

Assuming δ_1 and α are independent of n, one can obtain the sample complexity of AT-LUCB as follows: equate the second element in the max operator in (6) to a target failure rate δ and solve it for t. Using $n \le H^{\epsilon/2}$, this results in $O\left(H^{\epsilon/2}\max\left\{\log\left(\frac{H^{\epsilon/2}}{\delta}\right), \frac{\log^2(\frac{1}{\delta})}{\log(H^{\epsilon/2})}\right\}\right)$. Applying the same trick on the first element of the max results in $O\left(H^{\epsilon/2}\log\left(H^{\epsilon/2}/\delta\right)\right)$. We claim that the sample complexity of AT-LUCB is *practically*

$$O\left(H^{\epsilon/2}\log\left(H^{\epsilon/2}/\delta\right)\right)$$

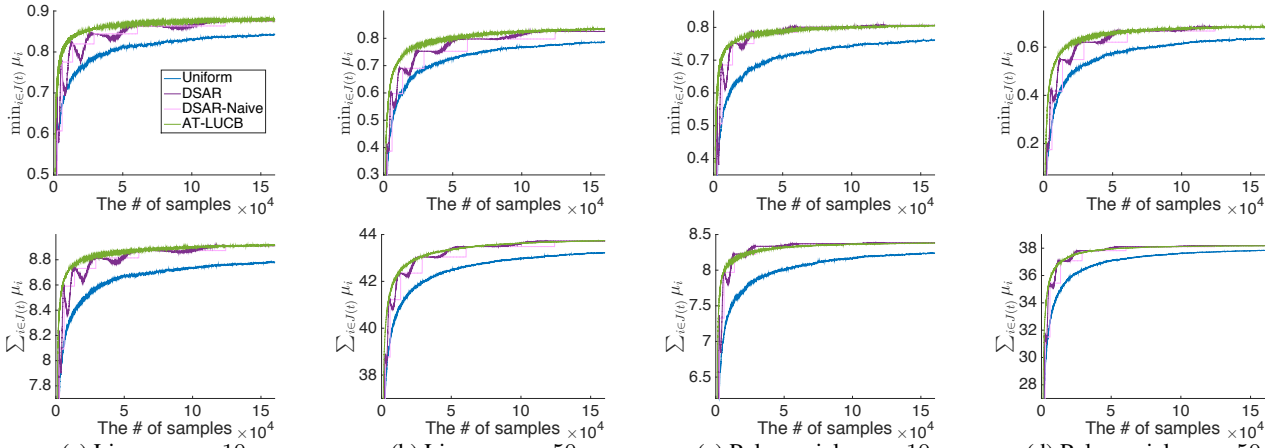
since the first element of the max operator in (6) dominates virtually everywhere as explained above.

Note the condition $t \ge T_1^*$ of Corollary 1 is not a disadvantage when compared to the guarantees of other algorithms such as doubling SAR since the probability bounds are all vacuous (> 1) until reaching a reasonably large t.

3. Experiments

We demonstrate the empirical performance of AT-LUCB by comparing it to the state-of-the-art baseline methods. All the following methods except **DSAR-Naive** recommends at each time step the empirical top-m arms based on *all* the samples collected so far.

- **Uniform**: we sample the least pulled arm at each time step.
- **DSAR**: we apply the doubling trick to SAR (Bubeck et al., 2013); for stage $s = 1, 2, \ldots$, run SAR with budget $2^{s-1}n$.



(a) Linear, $m = 10$ (b) Linear, $m = 50$ (c) Polynomial, $m = 10$ (d) Polynomial, $m = 50$

Figure 1. Anytime Explore-m results

- **DSAR-Naive**: a variant of **DSAR** where we recommend the empirical top-m from the end of the previous stage (motivation explained later).
- **AT-LUCB**: we run AT-LUCB with $\delta_1 = 1/2$, $\alpha = .99$, and $\epsilon = 0$.
- **UCB** (Bubeck et al., 2009) ($m = 1$ only): we set the exploration parameter α of UCB as 2.
- **DSH** ($m = 1$ only): we apply the doubling trick to SH (Karnin et al., 2013).

We omit UCB that recommends the most played arm (Bubeck et al., 2009) in experiments since it was always outperformed by UCB that recommends the empirical best arm.

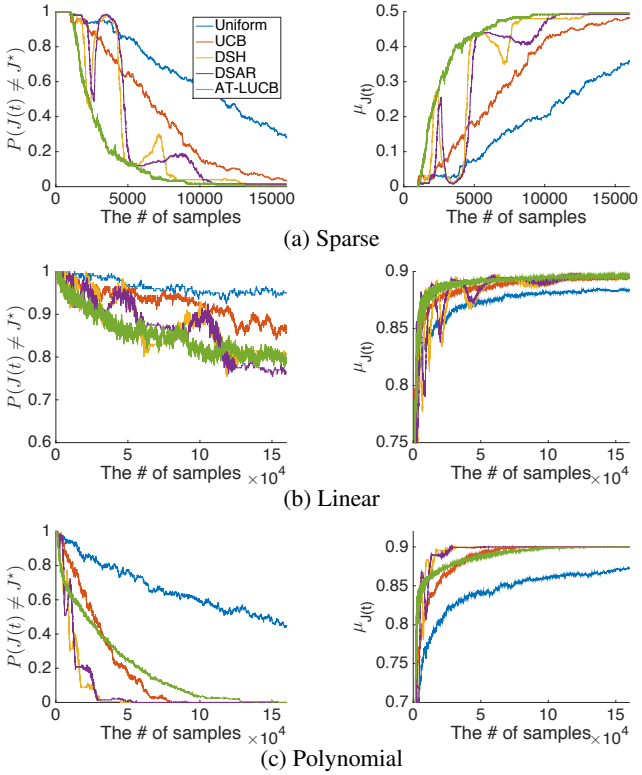
3.1. Toy MAB Instance

We consider toy MAB instances with $n = 1000$ where each arm is a Gaussian distribution with variance 1/4. Although the rewards are not necessarily in [0,1], the theoretical results of all the methods hold true. We consider two MAB instances: Linear and Polynomial. Linear increases its gap linearly: $\mu_i = .9\left(\frac{n-i}{n-1}\right), \forall i$. Polynomial increases its gaps polynomially: $\mu_1 = .9$ and $\mu_i = .9(1 - \sqrt{i/n}), \forall i \geq 2$.

We run each method 200 times with $m \in \{10, 50\}$. The misidentification probability $\mathbb{P}(J(t) \neq J^\star)$ on which algorithms are analyzed does not lead to a meaningful comparison since all are equally bad — it takes a long time to exactly find the top-m. Instead, we compare the smallest mean $\min_{i \in J(t)} \mu_i$ and the sum of the means $\sum_{i \in J(t)} \mu_i$ on Linear and Polynomial; see Figure 1. **AT-LUCB** outperforms both **Uniform** and **DSAR** overall. We observe a periodic performance fluctuation of **DSAR** in every experiment. Such a behavior stems from their elimination-based approach. For example, consider running **DSAR** with $m = 1$ and $n = 1000$ for ease of exposition. The first stage pulls each arm once. In the second stage (2000 budget), about 800 arms are pulled only once then eliminated. Suppose that at the end of the second stage the empirical best arm is truly the best one. At this point, the best arm is pulled about 73 times whereas 857 arms are pulled twice. In the third stage, the empirical means of those twice-pulled arms vary a lot when pulled, so we are likely to have one of them as the empirical best. At the end of the third stage, however, **DSAR** is likely to have the true best arm as the empirical best arm again since it keeps pulling the empirical best arm. In words, recommending the empirical top-m in the middle of a stage could be harmful. **DSAR-Naive** is a quick fix that prevents the fluctuation but is often worse than DSAR. While we found no easy fix[4], such a behavior is not found in methods based on confidence bound like AT-LUCB and UCB since they do not eliminate arms; the risk of less pulled arms being the empirical best is taken care of gradually. Note that in Polynomial **DSAR** performs slightly better than **AT-LUCB** in the end, but its unstable behavior in earlier stages makes it less attractive.

Anytime Explore-1 We run each method 200 times with $m = 1$. We estimate the misidentification probability $\mathbb{P}(J(t) \neq J^\star)$ and compute the mean of the recommended arm $\mu_{J(t)}$. We introduce a MAB instance called Sparse that has one and only one outstanding arm; $\mu_1 = .5$ and $\mu_i = 0, \forall i \geq 2$. The result is summarized in Figure 2. We omit **DSAR-Naive** for brevity. We make two observations. First, **AT-LUCB** performs better than all the other baselines in Sparse and Linear. We also observe the performance fluctuation from **DSH** just like **DSAR** due to its elimination-based approach. Second, **AT-LUCB** is outperformed by **DSAR** and **DSH** in Polynomial. The superiority of **DSAR** and **DSH** in Polynomial is attributed to the fact that their performance depends on $H_2^{\langle 1 \rangle}$ rather than $H^{\langle 1 \rangle}$. Recall that the sample complexity of **DSAR**

[4]We tried recommending the most pulled arm while breaking ties with empirical mean, but it worsened the overall performance.



(a) Sparse

(b) Linear

(c) Polynomial

Figure 2. Anytime Explore-1 results

is $O(H_2^{\langle m \rangle} \ln(n) \ln(n/\delta))$ and that $H_2^{\langle m \rangle} \leq H^{\langle m \rangle} \leq H_2^{\langle m \rangle} \ln(2n)$. Although $H_2^{\langle m \rangle}$ is close to $H^{\langle m \rangle}$ in general, $H_2^{\langle m \rangle}$ can be as small as $H^{\langle m \rangle} \log(2n)$. Indeed, $H^{\langle 1 \rangle}/H_2^{\langle 1 \rangle}$ is 1 for Sparse and 1.32 for Linear, but 6.99 for Polynomial. The same argument applies to **DSH**, too. Note that for $m \in \{10, 50\}$, $H^{\langle m \rangle}/H_2^{\langle m \rangle}$ of Polynomial drops below 1.6, which explains why we did not observe the superiority of **DSAR** in the previous experiments.

3.2. Application: Cartoon Caption Contest

The New Yorker cartoon caption contest[5] gives readers a chance to take their best shot at writing the funniest caption for a given cartoon. After receiving a large set of caption entries, the staffs have to sort through the entries to find the funniest one, which is a monumental task. The New Yorker recently started using a crowdsourcing system NEXT (Jamieson et al., 2015) to rate the degree of funniness of each caption entry with "not funny", "somewhat funny", or "funny". Each volunteers rates at most 25 captions. After one day, the staff selects the top-50 captions based on the average rating, and then performs a qualitative evaluation to choose the best one, which drastically saves the staffs' time and effort. A good crowdsourcing system should adaptively choose captions for rating so as to identify top-50 more quickly and accurately. The nature of the application is that the sampling budget (number of volunteers and the number of ratings from each) is limited and unknown. Anytime Explore-m is the right fit rather than the fixed budget setting that requires known budget or the fixed confidence setting that assumes one can obtain as many samples as necessary for the desired confidence. Presently, their crowdsourcing uses a fixed-confidence algorithm, such as lil'UCB (Jamieson et al., 2014), to choose which caption to rate next. We expect that AT-LUCB will be a better alternative.

[5] http://contest.newyorker.com/

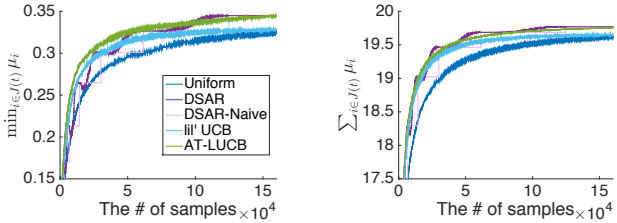


Figure 3. Results on cartoon caption contest data

We use the New Yorker dataset.[6] The data consists of $n = 496$ captions with ∼100K ratings. There were ∼4.5K volunteers where each rated ∼22 captions on average. Each caption received at least 100 ratings. We map "not funny" to 0, "somewhat funny" to $1/2$, and "funny" to 1. We then create a MAB instance where each caption's reward distribution is a multinomial over $\{0, 1/2, 1\}$ that is estimated by the MLE. We run each method 200 times on the synthesized MAB instance with $m = 50$. Figure 3 shows the anytime performance w.r.t. the smallest mean $\min_{i \in J(t)} \mu_i$ and the sum of the means $\sum_{i \in J(t)} \mu_i$. In terms of the smallest mean, **AT-LUCB** outperforms **DSAR** whereas in terms of the sum of the means **AT-LUCB** and **DSAR** are on par. In both cases, however, **DSAR** suffers from the performance oscillation under insufficient sample size. Again, there is no easy fix for this issue. We also run lil' UCB despite the mismatch in the setting (top-1, fixed-confidence) since it is currently being used by the system. lil' UCB performs worse than both **AT-LUCB** and **DSAR**, and even shows a similar performance to Uniform in the end.

4. Proof of Theorem 1

It suffices to prove (5). We introduce our notations. Let Arms $:= [1..n]$ be the set of n arms. Define $c = (\mu_m + \mu_{m+1})/2$, $\text{Above}^t(\delta) := \{a \in \text{Arms} : L_a^t(\delta) > c\}$, $\text{Below}^t(\delta) := \{a \in \text{Arms} : U_a^t(\delta) < c\}$, and $\text{Middle}^t(\delta) := \text{Arms} \setminus (\text{Above}^t(\delta) \cup \text{Below}^t(\delta))$. We define event

$$\text{Cross}_a^t(\delta) := \begin{cases} a \in \text{Below}^t(\delta) & \text{if } a \in [1..m] \\ a \in \text{Above}^t(\delta) & \text{if } a \in [m+1..n] \end{cases},$$

which means that the arm a is "disguising" to be a member of the opposite side, and

[6] Dataset number 499 from https://github.com/nextml/NEXT-data/.

$$\text{Needy}_a^t(\delta) := (a \in \text{Middle}^t(\delta)) \wedge \left(\beta(u_a^t, t, \delta) > \frac{\epsilon}{2}\right) ,$$

which means the arm a needs to be pulled.

The following lemma states that if the event $\text{Cross}_a^t(\delta_{S(t)})$ does not happen for any arm a, then at least one of the two arms chosen by AT-LUCB is in the middle and needs to be pulled. We present the proof in our supplementary material.

Lemma 1. *In AT-LUCB,*

$$\bigcap_{a\in \textit{Arms}} \neg \textit{Cross}_a^t(\delta_{S(t)}) \implies \textit{Needy}^t_{h_*^t(\delta_{S(t)})}(\delta_{S(t)}) \vee \textit{Needy}^t_{\ell_*^t(\delta_{S(t)})}(\delta_{S(t)})$$

Let $k_1 = 5/4$. Define

$$u^*(a,t,\delta) := \left\lceil \frac{1}{2\left(\Delta_a \vee \frac{\epsilon}{2}\right)^2} \ln\left(\frac{k_1 n t^4}{\delta}\right)\right\rceil ,$$

a sufficiently large number of samples of arm a such that $\beta(u^*(a,t,\delta),t,\delta)$ is no greater than $(\Delta_a \vee \frac{\epsilon}{2})$. The following lemma consists of three parts. Part *(i)* and *(ii)* state that the undesirable events are less likely to happen as t gets larger, which extends Lemma 3 and 4 of (Kalyanakrishnan et al., 2012) from the fixed δ to the random variable $\delta_{S(t)}$. Then, part *(iii)* states that AT-LUCB finishes stage s with sufficiently large probability if $t \geq T_s^*$: $\mathbb{P}(S(t) \leq s) \leq \frac{\delta_s}{10^4 \alpha \delta_1}$. This implies (5) using $\alpha \geq 1/50$ and $\delta_1 \geq 1/200$, which concludes the proof of Theorem 1.

Lemma 2. *Let $T_0^* := 1$. Under AT-LUCB1,*
(i) (No crossing arms) $\forall s \geq 1, \forall t \in [T_{s-1}^*..(T_s^*-1)]$,

$$\mathbb{P}(\cup_{a\in \textit{Arms}}\ \textit{Cross}_a^t(\delta_{S(t)})) \leq \frac{c_3\delta_s}{k_1 t^3}, \textit{where } c_3 = \frac{4}{3} . \quad (7)$$

(ii) (No sticky arms) $\forall s \geq 1, \forall t \in [T_{s-1}^*..(T_s^*-1)]$,

$$\mathbb{P}(\cup_{a\in \textit{Arms}}\ \{u_a^t > 4u^*(a,t,\delta_{S(t)})\} \wedge \textit{Needy}_a^t(\delta_{S(t)})) \leq \frac{c_3 H^{\epsilon/2}\delta_s}{k_1 n t^4}, \textit{where } c_3 = \frac{4}{3} . \quad (8)$$

(iii) For all stages $s \geq 2$ and for every time $t \in [T_{s-1}^*..(T_s^*-1)]$, the probability that AT-LUCB does not enter the stage s until time step t is at most δ_{s-1}:

$$\mathbb{P}(S(t) \leq s-1) \leq \frac{\delta_{s-1}}{10^4\alpha\delta_1}. \quad (9)$$

Proof. The proof of *(i)*, *(ii)*, and *(iii)* are interdependent. *(i)* and *(ii)* for $s = 1$ are implied by Lemma 3 and 4 of (Kalyanakrishnan et al., 2012). By induction, it suffices to assume that *(i)* and *(ii)* are true for stage $s \in [1..d]$ and prove *(iii)* for $s = d+1$ (step 1) and *(i)* and *(ii)* for $s = d+1$ (step 2). Now, assume *(i)* and *(ii)* are true for stage $s \in [1..d]$ with $d \geq 1$.

Step 1: prove *(iii)* for $s = d+1$.
The proof of the step 1 extends the proof of Lemma 5 in (Kalyanakrishnan et al., 2012) to the case where the failure rate δ used by the algorithm is a random variable $\delta_{S(t)}$.

Let $t \in [T_d^*..(T_{d+1}^*-1)]$. Let $\bar{t} = \lceil \gamma_d^* t \rceil$ where γ_d^* is defined in (3). Define two events $E_1(\tau)$ and $E_2(\tau)$:

$$E_1(\tau) = \cup_{a\in\text{Arms}}\ \text{Cross}_a^\tau(\delta_{S(\tau)})$$
$$E_2(\tau) = \cup_{a\in\text{Arms}}\ \{u_a^\tau > 4u^*(a,\tau,\delta_{S(\tau)})\} \wedge \text{Needy}_a^\tau(\delta_{S(\tau)})$$

Step 1-1: show that if $\bigcap_{\tau=\bar{t}}^{t-1}\{\neg E_1(\tau) \wedge \neg E_2(\tau)\}$ then AT-LUCB must have finished stage d at time t ($S(t) \geq d+1$).

Suppose that the algorithm has not entered stage $d+1$ after sampling $\bar{t}-1$ times ($S(\bar{t}) \leq d$) since if it has, then there is nothing left to prove. Consider the case where there were no cross or sticky arms at time $\bar{t}, \ldots, (t-1)$: $\cap_{\tau=\bar{t}}^{t-1}(\neg E_1(\tau) \wedge \neg E_2(\tau))$. Let $\#steps$ be the number of additional number of time steps (twice the number of samples) before entering stage $d+1$. Let $A_*(\tau,\delta) := \{h_*^\tau(\delta), \ell_*^\tau(\delta)\}$. Then, by the assumption $\cap_{\tau=\bar{t}}^{t-1}\neg E_1(\delta_{S(\tau)})$,

$$\begin{aligned}
&\#steps \\
&= \sum_{\tau=\bar{t}}^{t-1} \mathbb{1}\left\{(S(\tau) \leq d) \wedge (\cap_{a\in\text{Arms}} \neg\text{Cross}_a^\tau(\delta_{S(\tau)}))\right\} \\
&\overset{(a)}{\leq} \sum_{\tau=\bar{t}}^{t-1} \mathbb{1}\big\{(S(\tau) \leq d) \wedge \\
&\quad \left(\text{Needy}^\tau_{h_*^\tau(\delta_{S(\tau)})}(\delta_{S(\tau)}) \vee \text{Needy}^\tau_{\ell_*^\tau(\delta_{S(\tau)})}(\delta_{S(\tau)})\right)\big\} \\
&\leq \sum_{\tau=\bar{t}}^{t-1} \sum_{a\in\text{Arms}} \mathbb{1}\big\{(S(\tau) \leq d) \wedge \left(a \in A_*(\tau,\delta_{S(\tau)})\right) \wedge \\
&\qquad \text{Needy}_a^\tau(\delta_{S(\tau)})\big\} \\
&\overset{(b)}{\leq} \sum_{\tau=\bar{t}}^{t-1} \sum_{a\in\text{Arms}} \mathbb{1}\big\{(S(\tau) \leq d) \wedge \left(a \in A_*(\tau,\delta_{S(\tau)})\right) \wedge \\
&\qquad (u_a^\tau \leq 4u^*(a,\tau,\delta_{S(\tau)}))\big\} \\
&\leq \sum_{\tau=\bar{t}}^{t-1} \sum_{a\in\text{Arms}} \mathbb{1}\big\{(S(\tau) \leq d) \wedge \left(a \in A_*(\tau,\delta_{S(\tau)})\right) \wedge \\
&\qquad (u_a^\tau \leq 4u^*(a,t,\delta_d))\big\} \\
&\leq \sum_{\tau=\bar{t}}^{t-1} \sum_{a\in\text{Arms}} \mathbb{1}\left\{u_a^\tau \leq 4u^*(a,t,\delta_d)\right\} \\
&\leq \sum_{a\in\text{Arms}} 4u^*(a,t,\delta_d),
\end{aligned}$$

where (a) is due to Lemma 1 and (b) is due to $\bigcap_{\tau=\bar{t}}^{t-1}\neg E_2(\tau)$. Then, $\bar{t} - 1 + \#steps \leq \lceil\gamma_d^* t\rceil - 1 + \sum_a 4u^*(a,t,\delta_d) < t$, where the last inequality is by Lemma 4 in our supplementary material. This concludes step 1-1.

Step 1-2: show $\mathbb{P}\left(\bigcup_{\tau=\bar{t}}^{t-1} E_1(\tau) \vee E_2(\tau)\right) \leq \frac{\delta_d}{10^4\alpha\delta_1}$.

Define $x^* := \log_{1/\alpha} \frac{12k_1nH^{\epsilon/2}\sqrt{e}}{\delta_1}$. We prove the claim in two cases: $d \le \lceil x^* \rceil$ and $d \ge \lceil x^* \rceil + 1$.

Case (a): $d \le \lceil x^* \rceil$
Using the union bound and $\bar{t} \ge \gamma_d^* t$,

$$\begin{aligned}
&\mathbb{P}\left(\bigcup_{\tau=\bar{t}}^{t-1} E_1(\tau) \vee E_2(\tau)\right) \\
&\le \sum_{\tau=\bar{t}}^{t-1} \left(\frac{c_3\delta_1}{k_1\tau^3} + \frac{c_3H^{\epsilon/2}\delta_1}{k_1n\tau^4}\right) \text{, by (7) and (8) with } s=1 \\
&\le (1-\gamma_d^*)t\left(\frac{c_3\delta_1}{k_1\bar{t}^3} + \frac{c_3H^{\epsilon/2}\delta_1}{k_1n\bar{t}^4}\right) \\
&\le (1-\gamma_d^*)\frac{c_3}{k_1(\gamma_d^*)^3t^2}\left(1 + \frac{H^{\epsilon/2}}{n\gamma_d^*t}\right)\frac{\delta_1}{\delta_d}\delta_d \,.
\end{aligned}$$

Case (b): $d \ge \lceil x^* \rceil + 1$
Lemma 5 in our supplementary material shows that $T^*_{d-\lceil x^* \rceil} \le \lceil \gamma_d^* T_d^* \rceil$, which leads to $\bar{t} = \lceil \gamma_d^* t \rceil \ge \lceil \gamma_d^* T_d^* \rceil \ge T^*_{d-\lceil x^* \rceil} \ge T^*_{d-\lceil x^* \rceil - 1}$. Then, (7) and (8) with $s = d - \lceil x^* \rceil$ imply that

$$\begin{aligned}
&\mathbb{P}\left(\bigcup_{\tau=\bar{t}}^{t-1} E_1(\tau) \vee E_2(\tau)\right) \\
&\le \sum_{\tau=\bar{t}}^{t-1} \left(\frac{c_3\delta_{d-\lceil x^* \rceil}}{k_1\tau^3} + \frac{c_3H^{\epsilon/2}\delta_{d-\lceil x^* \rceil}}{k_1n\tau^4}\right) \\
&\le (1-\gamma_d^*)\frac{c_3}{k_1(\gamma_d^*)^3t^2}\left(1 + \frac{H^{\epsilon/2}}{n\gamma_d^*t}\right)\frac{\delta_{d-\lceil x^* \rceil}}{\delta_d}\delta_d \,.
\end{aligned}$$

Lemma 6 in our supplementary material shows that both $\frac{\delta_1}{\delta_d}$ and $\frac{\delta_{d-\lceil x^* \rceil}}{\delta_d}$ above are no greater than $\frac{12k_1nH^{\epsilon/2}\sqrt{e}}{\alpha\delta_1}$. This implies

$$\begin{aligned}
&\mathbb{P}\left(\bigcup_{\tau=\bar{t}}^{t-1} E_1(\tau) \vee E_2(\tau)\right) \\
&\le (1-\gamma_d^*)\frac{c_3}{k_1(\gamma_d^*)^3t^2}\left(1 + \frac{H^{\epsilon/2}}{n\gamma_d^*t}\right)\frac{12k_1nH^{\epsilon/2}\sqrt{e}}{\alpha\delta_1}\delta_d \\
&\le \frac{16}{27}\frac{c_3}{t^2}\left(1 + \frac{H^{\epsilon/2}}{n\gamma_d^*t}\right)\frac{12nH^{\epsilon/2}\sqrt{e}}{\alpha\delta_1}\delta_d \text{, since } \gamma_d^* \ge 3/4 \\
&\le \frac{16}{27}c_3\left(1 + \frac{H^{\epsilon/2}}{n\gamma_d^*T_d^*}\right)\frac{12nH^{\epsilon/2}\sqrt{e}}{(T_d^*)^2\alpha\delta_1}\delta_d \,.
\end{aligned}$$

Note that, since $\gamma_d^* \ge 3/4$, $\delta_d \le \delta_1 \le n$ and $k_1 = 5/4$,

$$T_d^* \ge \frac{24H^{\epsilon/2}}{1/4}\ln\left(\frac{12k_1nH^{\epsilon/2}}{(1/4)\delta_d}\right) \ge 96H^{\epsilon/2}\ln(60H^{\epsilon/2}) \,.$$

Then,

$$\begin{aligned}
\frac{H^{\epsilon/2}}{n\gamma_d^*T_d^*} &\le \frac{H^{\epsilon/2}}{n(3/4)96H^{\epsilon/2}\ln(60H^{\epsilon/2})} \\
&\le \frac{H^{\epsilon/2}}{3(3/4)96H^{\epsilon/2}\ln(60\cdot 3)} \text{, since } H^{\epsilon/2} \ge n \ge 3 \\
&< \tfrac{1}{1080} \text{, since } \ln(180) > 5,
\end{aligned}$$

and with a similar reasoning

$$\begin{aligned}
\frac{12nH^{\epsilon/2}\sqrt{e}}{(T_d^*)^2\alpha\delta_1} &\le \frac{12nH^{\epsilon/2}\sqrt{e}}{96^2(H^{\epsilon/2})^2\ln^2(60H^{\epsilon/2})\alpha\delta_1} \\
&\le \frac{12nH^{\epsilon/2}2}{96^2(H^{\epsilon/2})^2\ln^2(180)\alpha\delta_1} \\
&< \frac{12\cdot 2}{96^2 5^2\alpha\delta_1} = \frac{1}{9600\alpha\delta_1} \,.
\end{aligned}$$

Finally, using $c_3 = 4/3$,

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{\tau=\bar{t}}^{t-1} E_1(\tau) \vee E_2(\tau)\right) &\le \frac{16}{27}c_3\frac{1081}{1080}\frac{1}{9600\alpha\delta_1}\delta_d \\
&\le \frac{\delta_d}{10^4\alpha\delta_1} \,.
\end{aligned}$$

Step 2: prove *(i)* and *(ii)* for $s = d+1$.
Let $t \in [T_d^* .. T_{d+1}^* - 1]$. Define $\text{Cross}^t(\delta) := \cup_{a\in\text{Arms}} \text{Cross}_a^t(\delta)$. It is not hard to see that, using Lemma 7,

$$\mathbb{P}(\text{Cross}^t(\delta_{S(t)}) \mid S(t) \ge d+1) \le \frac{\delta_{d+1}}{k_1t^3},$$

where we emphasize that the conditioning part is $S(t) \ge d+1$ rather than $S(t) = d+1$. Similarly, $\mathbb{P}(\text{Cross}^t(\delta_{S(t)}) \mid S(t) \le d) \le \frac{\delta_1}{k_1t^3}$. Then,

$$\begin{aligned}
&\mathbb{P}(\text{Cross}^t(\delta_{S(t)})) \\
&= \mathbb{P}(S(t) \ge d+1)\mathbb{P}(\text{Cross}^t(\delta_{S(t)}) \mid S(t) \ge d+1) + \\
&\quad \mathbb{P}(S(t) \le d)\mathbb{P}(\text{Cross}^t(\delta_{S(t)}) \mid S(t) \le d) \\
&\le 1 \cdot \frac{\delta_{d+1}}{k_1t^3} + \frac{\delta_d}{10^4\alpha\delta_1}\cdot\frac{\delta_1}{k_1t^3} \text{, since the step 1} \\
&= \frac{\delta_{d+1}}{k_1t^3}\left(1 + \frac{1}{10^4\alpha^2\delta_1}\delta_1\right) \text{, since } \tfrac{\delta_d}{\delta_{d+1}} = \tfrac{1}{\alpha} \\
&\le \frac{5}{4}\frac{\delta_{d+1}}{k_1t^3} < \frac{4}{3}\frac{\delta_{d+1}}{k_1t^3} \text{, since } \alpha \ge 1/50 \,.
\end{aligned}$$

Similarly, define $\text{Sticky}^t(\delta) := \cup_{a\in\text{Arms}} \left\{u_a^\tau > 4u^*(a,t,\delta_{S(\tau)})\right\} \wedge \text{Needy}_a^\tau(\delta_{S(\tau)})$.

$$\begin{aligned}
&\mathbb{P}(\text{Sticky}^t(\delta_{S(t)})) \\
&= \mathbb{P}(S(t) \ge d+1)\mathbb{P}(\text{Sticky}^t(\delta_{S(t)}) \mid S(t) \ge d+1) + \\
&\quad \mathbb{P}(S(t) \le d)\mathbb{P}(\text{Sticky}^t(\delta_{S(t)}) \mid S(t) \le d) \\
&\le 1 \cdot \frac{3H^{\epsilon/2}\delta_{d+1}}{4k_1nt^4} + \frac{\delta_d}{10^4\alpha\delta_1}\cdot\frac{3H^{\epsilon/2}\delta_1}{4k_1nt^4} \\
&\le \frac{3H^{\epsilon/2}\delta_{d+1}}{4k_1nt^4}\left(1 + \frac{1}{10^4\alpha^2\delta_1}\delta_1\right) \\
&\le \frac{15}{16}\frac{H^{\epsilon/2}\delta_{d+1}}{k_1nt^4} < \frac{4}{3}\frac{H^{\epsilon/2}\delta_{d+1}}{k_1nt^4}
\end{aligned}$$

□

Acknowledgements

The authors thank Kevin Jamieson for helpful feedback on this work and the anonymous reviewers for the comments. This work was partially supported by NSF grant CCF-1218189 and NIH grant 1 U54 AI117924-01.

References

Audibert, Jean-Yves, Bubeck, Sébastien, and Munos, Rémi. Best arm identification in multi-armed bandits. In *Proceedings of the Conference on Learning Theory (COLT)*, 2010.

Bechhofer, Robert E. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. *Biometrics*, 14(3): 408–429, 1958.

Bubeck, Sébastien, Munos, Rémi, and Stoltz, Gilles. Pure exploration in multi-armed bandits problems. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, pp. 23–37, 2009.

Bubeck, Sébastien, Wang, Tengyao, and Viswanathan, Nitin. Multiple identifications in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 258–265, 2013.

Even-dar, Eyal, Mannor, Shie, and Mansour, Yishay. Pac bounds for multi-armed bandit and markov decision processes. In *Proceedings of the Conference on Learning Theory (COLT)*, pp. 255–270, 2002.

Jamieson, Kevin, Malloy, Matthew, Nowak, Robert, and Bubeck, Sébastien. lil' ucb : An optimal exploration algorithm for multi-armed bandits. In *Proceedings of the Conference on Learning Theory (COLT)*, pp. 423–439, 2014.

Jamieson, Kevin, Jain, Lalit, Fernandez, Chris, Glattard, Nicholas, and Nowak, Robert. Next: A system for real-world development, evaluation, and application of active learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Kalyanakrishnan, Shivaram, Tewari, Ambuj, Auer, Peter, and Stone, Peter. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 655–662, 2012.

Karnin, Zohar Shay, Koren, Tomer, and Somekh, Oren. Almost optimal exploration in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1238–1246, 2013.

Kaufmann, Emilie and Kalyanakrishnan, Shivaram. Information complexity in bandit subset selection. In *Proceedings of the Conference on Learning Theory (COLT)*, pp. 228–251, 2013.

Paulson, Edward. A sequential procedure for selecting the population with the largest mean from k normal populations. *Annals of Mathematical Statistics*, 35(1):174–180, 1964.