In Langford, Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, pp. 655--662, Omnipress, New York, NY, USA, 2012.

PAC Subset Selection in Stochastic Multi-armed Bandits

Shivaram Kalyanakrishnan[1] SHIVARAM@YAHOO-INC.COM
Ambuj Tewari[2] AMBUJ@CS.UTEXAS.EDU
Peter Auer[3] AUER@UNILEOBEN.AC.AT
Peter Stone[2] PSTONE@CS.UTEXAS.EDU
[1]Yahoo! Labs Bangalore, Bengaluru Karnataka 560071 India
[2]Department of Computer Science, The University of Texas at Austin, Austin Texas 78701 USA
[3]Chair for Information Technology, University of Leoben, Leoben 8700 Austria

Abstract

We consider the problem of selecting, from among the arms of a stochastic n-armed bandit, a subset of size m of those arms with the highest expected rewards, based on efficiently sampling the arms. This "subset selection" problem finds application in a variety of areas. In the authors' previous work (Kalyanakrishnan & Stone, 2010), this problem is framed under a PAC setting (denoted "Explore-m"), and corresponding sampling algorithms are analyzed. Whereas the formal analysis therein is restricted to the worst case sample complexity of algorithms, in this paper, we design and analyze an algorithm ("LUCB") with improved expected sample complexity. Interestingly LUCB bears a close resemblance to the well-known UCB algorithm for regret minimization. The expected sample complexity bound we show for LUCB is novel even for single-arm selection (Explore-1). We also give a lower bound on the worst case sample complexity of PAC algorithms for Explore-m.

1. Introduction

We consider the Explore-m problem introduced previously by the authors (Kalyanakrishnan & Stone, 2010). The problem is that of selecting, from among the arms of a stochastic n-armed bandit, a subset of size m of those arms with the highest expected rewards, based on efficiently sampling the arms. This subset selection problem finds application in a variety of areas, such as simulation, industrial engineering, on-line advertising, and also within certain stochastic optimization techniques. Explore-m generalizes Explore-1, a PAC setting introduced by Even-Dar et al. (2006) for selecting the (single) best arm.

The authors' previous work introduces the Halving algorithm for Explore-m (Kalyanakrishnan & Stone, 2010). While this algorithm improves upon the sample complexity of a uniform sampling strategy, its sample complexity is identical across different bandit instances—therefore not implementing the intuition that in bandit instances where the highest m and the lowest $n - m$ means of the arms are separated by a relatively large margin, fewer samples should suffice for reliably identifying the best arms. In other words, while the Halving algorithm is *sufficient* for achieving a PAC guarantee in the *worst* case, we are apt to wonder if on "easier" bandit instances, a sampling algorithm can finish any earlier. The question we confront is akin to one addressed by Schuurmans & Greiner (1995), who show in the context of supervised learning that while preserving a PAC correctness guarantee over an entire class of problems, often it is possible to improve efficiency on a given problem instance by sequentially adapting based on the samples observed.

As the first contribution of this paper, we present the LUCB algorithm for Explore-m, which meets the relevant PAC requirement, and enjoys an expected sample complexity bound that increases with a natural measure of problem complexity. The algorithm maintains a clear separation between its stopping rule and sampling strategy, therein contrasting with previous elimination-based approaches for exploration (Even-Dar et al., 2006; Mnih et al., 2008). In elimination-based algorithms, it becomes difficult to ensure low sample complexity on runs in which erroneous eliminations occur. By contrast, LUCB deter-

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

mines the entire set of m arms to select (or $n - m$ to reject) only at termination. Consequently we can analyze LUCB's sample complexity separately from its correctness, and guarantee a low expected sample complexity (including runs on which mistakes are made). We believe such a bound is novel even for the single-arm case, and that the confidence bounds-based sampling approach of LUCB—again novel for a PAC setting—is a natural means for achieving the bound. Note that the worst case sample complexity of LUCB can be kept to within a constant factor of HALVING's.[1]

The second contribution of this paper is indeed a lower bound on the worst case sample complexity of PAC algorithms for EXPLORE-m. This lower bound shows that the worst case sample complexity of the HALVING algorithm is optimal up to a constant factor.

In Section 2, we review EXPLORE-m and introduce relevant terminology. In Section 3, we describe and analyze the LUCB algorithm. Section 4 presents our lower bound, and Section 5 serves as the conclusion.

2. Problem Setting and Terminology

Below we review the EXPLORE-m problem (Kalyanakrishnan & Stone, 2010) and introduce terms related to problem complexity.

Stochastic multi-armed bandits. We consider an arbitrary instance of an n-armed bandit, $n \geq 2$; let its arms be numbered $1, 2, \ldots, n$. Each sample (or "pull") of arm a yields a reward of either 0 or 1, generated randomly from a fixed Bernoulli distribution with mean $p_a \in [0, 1]$.[2] Indeed each bandit instance is completely determined by the distributions corresponding to its arms. For simplicity of notation we assume an indexing of the arms such that

$$p_1 \geq p_2 \geq \ldots \geq p_n. \tag{1}$$

Naturally the learner is unaware of this indexing. The random variables generating rewards for the arms are mutually independent. Arm a is defined to be (ϵ, m)-optimal, $\forall \epsilon \in (0, 1)$, $\forall m \in \{1, 2, \ldots, n-1\}$, iff

$$p_a \geq p_m - \epsilon. \tag{2}$$

[1] The ADAPT algorithm presented in the authors' previous work (Kalyanakrishnan & Stone, 2010) does not have a PAC guarantee as claimed therein. The authors regret the erroneous claim, and thank Gergely Neu for bringing it to their attention. LUCB was conceived to be a provably-correct and provably-efficient replacement for ADAPT.

[2] Hoeffding's inequality is the sole concentration bound used in the analysis of LUCB (Section 3): therefore, it is trivial to extend the algorithm and its analysis to bandits having reward distributions with known, bounded ranges.

We find it convenient to denote as *Arms* the set of all arms in our n-armed bandit instance; by *Top* the m arms with the highest mean rewards; and by *Bot* the $n - m$ arms with the lowest mean rewards.

$$\begin{aligned} Arms &\stackrel{\text{def}}{=} \{1, 2, \ldots, n\}, \\ Top &\stackrel{\text{def}}{=} \{1, 2, \ldots, m\}, \text{ and} \\ Bot &\stackrel{\text{def}}{=} \{m+1, m+2, \ldots, n\}. \end{aligned}$$

We see from (1) and (2) that every arm in *Top* is (ϵ, m)-optimal. Hence, there are *at least* m (ϵ, m)-optimal arms. Let *Good* be the set of all (ϵ, m)-optimal arms, and let the set *Bad* contain all the remaining arms. In general: $m \leq |Good| \leq n$ and $0 \leq |Bad| \leq (n - m)$.

EXPLORE-m problem. A bandit instance may be sampled sequentially in rounds. On each round $t \in \{1, 2, \ldots\}$, an algorithm for EXPLORE-m must either (1) select an arm a^t to sample, or (2) terminate and return an m-sized subset of *Arms*. The outcome of each sampling is a pair of the form (arm, reward), where the reward is drawn as an independent sample from the distribution associated with the arm. We refer to the sequence of outcomes up to (and *excluding*) round t as the *history* at round t: during each round t, this history is available to the algorithm. Note that we do not require the m arms returned by the algorithm to be in any particular order.

For $\delta \in (0, 1)$, an algorithm $\mathcal{A}$ is defined to be (ϵ, m, δ)-optimal, iff for every bandit instance: (1) with probability 1, $\mathcal{A}$ terminates in a finite number of rounds, and (2) with probability at least $1 - \delta$, every arm returned by $\mathcal{A}$ is (ϵ, m)-optimal. Note that EXPLORE-1, for which $(\epsilon, 1, \delta)$-optimal algorithms are to be designed, matches the formulation of Even-Dar et al. (2006).

The *sample complexity* of algorithm $\mathcal{A}$ during a terminating run is the total number of pulls it performs before termination. Assume that with probability 1, $\mathcal{A}$ indeed terminates in a finite number of rounds. Then, for a given bandit instance $\mathcal{B}$, the worst case sample complexity of $\mathcal{A}$ is the maximum sample complexity among its runs on $\mathcal{B}$, and the expected sample complexity the average over all its runs on $\mathcal{B}$. The (overall) worst case sample complexity of an algorithm is its maximum worst case sample complexity across all bandit instances.

Problem complexity. Before proceeding, we define quantities to describe the complexity of the EXPLORE-m problem for a bandit instance. Intuition suggests that the problem must be easier if the mean rewards of the arms are more separated. We denote by $\Delta_{i,j}$ the separation between the means of arms i and j:

$$\Delta_{i,j} \stackrel{\text{def}}{=} p_i - p_j.$$

We find it convenient to use additional shorthand for the separation of arms in Top from arm $m+1$, and the separation of arms in Bot from arm m.

$$\Delta_a \stackrel{\text{def}}{=} \begin{cases} \Delta_{a,m+1} & \text{if } 1 \leq a \leq m, \\ \Delta_{m,a} & \text{if } m+1 \leq a \leq n. \end{cases}$$

Observe that $\Delta_m = \Delta_{m+1} = p_m - p_{m+1}$. Let c denote the mid-point of the means of arms m and $m+1$; it is easy to establish a relationship between p_a, Δ_a, and c.

$$c \stackrel{\text{def}}{=} \frac{p_m + p_{m+1}}{2}.$$

$$\forall a \in Arms : \frac{\Delta_a}{2} \leq |p_a - c| \leq \Delta_a. \tag{3}$$

For $\gamma \in [0,1]$, we denote the larger of Δ_a and γ as $[\Delta_a \vee \gamma]$, and aggregate over all the arms to derive a complexity measure H^γ.

$$[\Delta_a \vee \gamma] \stackrel{\text{def}}{=} \max\{\Delta_a, \gamma\}.$$

$$H^\gamma \stackrel{\text{def}}{=} \sum_{a \in Arms} \frac{1}{[\Delta_a \vee \gamma]^2}.$$

Theorem 6 will show that the expected sample complexity of LUCB1 (an instance of LUCB) is $O\left(H^{\epsilon/2} \log\left(\frac{H^{\epsilon/2}}{\delta}\right)\right)$. Thus, the main attribute of LUCB1 is its improved expected sample complexity on bandit instances for which $H^{\epsilon/2}$ is relatively small (in practice we expect to mostly encounter such instances). By contrast, the HALVING algorithm uniformly has a sample complexity of $O\left(\frac{n}{\epsilon^2} \log\left(\frac{m}{\delta}\right)\right)$ over all bandit instances: therefore it is inefficient except on those instances for which $H^{\epsilon/2} \approx \frac{n}{\epsilon^2}$.

3. LUCB Algorithm

The algorithm we introduce in this paper for subset selection naturally decomposes into two elements: (1) a stopping rule that maps the set of histories to the set $\{\text{STOP}, \text{CONTINUE}\}$, and (2) a sampling strategy that maps the set of histories to $Arms$. The algorithm is initialized by sampling each arm once. On every subsequent round, the algorithm evaluates the stopping rule. If it must stop, it returns the m-sized subset of arms with the highest empirical means (assume some fixed rule exists for breaking ties). Otherwise it pulls an arm according to the sampling strategy, and continues. We name our algorithm "LUCB" based on the centrality of lower and upper confidence bounds in its stopping rule and sampling strategy.

3.1. Stopping Rule

The stopping rule under LUCB is of an intuitive form. During round t, let u_a^t denote the number of times arm a has been sampled, and let $\hat{p}_a^t$ be the empirical mean of the rewards from arm a. The key element of our stopping rule is a confidence bound $\beta(u_a^t, t)$, which is a positive number interpreted to be a high-probability bound on the deviation of the empirical mean of arm a from its true mean. In particular the lower confidence bound for arm a during round t is given by $\hat{p}_a^t - \beta(u_a^t, t)$, and the upper confidence bound is given by $\hat{p}_a^t + \beta(u_a^t, t)$.

During round t, let $High^t$ be the set of m arms with the highest empirical averages, and Low^t be the set of $n-m$ arms with the lowest empirical averages. Among the arms in $High^t$, let h_*^t be the arm with the lowest lower confidence bound; among the arms in Low^t, let l_*^t be the arm with the highest upper confidence bound:

$$h_*^t \stackrel{\text{def}}{=} \operatorname*{argmin}_{h \in High^t} \{\hat{p}_h^t - \beta(u_h^t, t)\}, \text{ and}$$

$$l_*^t \stackrel{\text{def}}{=} \operatorname*{argmax}_{l \in Low^t} \{\hat{p}_l^t + \beta(u_l^t, t)\}.$$

Our stopping rule is to terminate iff

$$\boxed{\left(\hat{p}_{l_*^t}^t + \beta(u_{l_*^t}^t, t)\right) - \left(\hat{p}_{h_*^t}^t - \beta(u_{h_*^t}^t, t)\right) < \epsilon.}$$

If the algorithm does stop, then $High^t$ is returned. We show that if $\beta(u,t)$ is sufficiently large, then LUCB guarantees a low mistake probability (Section 3.3 provides a concrete choice of β).

Theorem 1. *Let $\mathcal{S}$: Set of histories $\rightarrow$ Arms be an arbitrary sampling strategy, and let $\beta : \{1,2,3,\dots\} \times \{1,2,3,\dots\} \rightarrow (0,\infty)$ be a function such that*

$$\sum_{t=1}^{\infty} \sum_{u=1}^{t} \exp\left(-2u\beta(u,t)^2\right) \leq \frac{\delta}{n}. \tag{4}$$

Then, if LUCB(n, m, ϵ, δ, $\mathcal{S}$, β) terminates, the probability that it returns a non-(ϵ, m)-optimal arm (an arm in Bad) is at most δ.

Proof. We employ a standard argument using confidence bounds. If indeed some arm b in Bad is returned, then LUCB must have terminated during some round t such that b is present in $High^t$ (and by implication, some arm i in Top is present in Low^t). Since $p_i - p_b > \epsilon$, from the termination rule, we infer that $\hat{p}_b^t > p_b + \beta(u_b^t, t)$ or $\hat{p}_i^t < p_i - \beta(u_i^t, t)$: that is, b or i has violated its upper or lower confidence bound, respectively. Hence, the algorithm's mistake probability is upper-bounded by the probability of the event that there exist t, b, i, u_b^t, and u_i^t such that $\hat{p}_b^t > p_b + \beta(u_b^t, t)$ or $\hat{p}_i^t < p_i - \beta(u_i^t, t)$. Regardless of the sampling strategy, note that u_b^t and u_i^t have to be between 1 and $t-1$. Applying Hoeffding's inequality along with the union bound and (4), we obtain the desired result. $\square$

3.2. Greedy Sampling Strategy

Whereas Theorem 1 applies to every sampling strategy, we show that it is *economical* to use a sampling strategy that is greedy with respect to the stopping rule. In particular during round t, the arms h^*_t and l^*_t can be construed as the arms that are most likely to lead to a mistake: naturally it would then be advisable to sample these arms instead of others. Implementing this very intuition, our sampling strategy is as follows:[3]

During round t: sample arms h^t_* and l^t_*.

Below we introduce some infrastructure for our analysis. Then, in Lemma 2, we establish that indeed h^t_* and l^t_* are good candidates for sampling. Subsequently in Section 3.3, we concretely pick a function, β_1, as a confidence bound, and formally bound the expected sample complexity of the resulting algorithm, which we denote LUCB1.

Recall that $c \stackrel{\text{def}}{=} \frac{p_m + p_{m+1}}{2}$. During round t, let us partition the set of arms into three sets: $Above^t$, which comprises arms whose lower confidence bounds fall above c; $Below^t$, which comprises arms whose upper confidence bounds fall below c; and $Middle^t$, the remainder.

$$\begin{aligned} Above^t &\stackrel{\text{def}}{=} \{a \in Arms : \hat{p}^t_a - \beta(u^t_a, t) > c\}. \\ Below^t &\stackrel{\text{def}}{=} \{a \in Arms : \hat{p}^t_a + \beta(u^t_a, t) < c\}. \\ Middle^t &\stackrel{\text{def}}{=} Arms \setminus \left(Above^t \cup Below^t\right). \end{aligned}$$

We expect that by and large, arms in Top will be in $Above^t$ or $Middle^t$, while arms in Bot will be in $Below^t$ or $Middle^t$. Let $CROSS^t_a$ denote the event that arm a does *not* obey such an expectation (and let $CROSS^t$ denote that some arm has "crossed").

$$\begin{aligned} CROSS^t_a &\stackrel{\text{def}}{=} \begin{cases} a \in Below^t & \text{if } a \in Top, \\ a \in Above^t & \text{if } a \in Bot. \end{cases} \\ CROSS^t &\stackrel{\text{def}}{=} \exists a \in Arms : CROSS^t_a. \end{aligned}$$

Now, let us define a "needy" arm as one in $Middle^t$ with a confidence bound greater than $\frac{\epsilon}{2}$: let $NEEDY^t_a$ be the event that arm a is needy during round t.

$$NEEDY^t_a \stackrel{\text{def}}{=} \left(a \in Middle^t\right) \wedge \left(\beta\left(u^t_a, t\right) > \frac{\epsilon}{2}\right).$$

Additionally let $TERM^t$ denote the event that during round t, the stopping rule will lead to termination:

$$TERM^t \stackrel{\text{def}}{=} \left(\hat{p}^t_{l^t_*} + \beta\left(u_{l^t_*}, t\right)\right) - \left(\hat{p}^t_{h^t_*} - \beta\left(u_{h^t_*}, t\right)\right) < \epsilon.$$

[3]Observe that we sample *two* arms every round (from round $n+1$ onwards). Thus, the sample complexity of our algorithm is at most twice the number of rounds.

The following key lemma shows that if $CROSS^t$ does not occur, and LUCB does not terminate during round t, then either h^t_* or l^t_* is a needy arm.

Lemma 2. $\neg CROSS^t \wedge \neg TERM^t \implies NEEDY^t_{h^t_*} \vee NEEDY^t_{l^t_*}.$

Proof. In our proof below, we reduce notational clutter by dropping the suffix t in our variables. Additionally we use the shorthand $\beta[a]$ for $\beta(u^t_a, t)$. To prove the lemma, we prove the following statements.

$$\neg CROSS \wedge \neg TERM \implies (h_* \in Middle) \vee (l_* \in Middle). \tag{5}$$

$$\neg TERM \wedge (h_* \in Middle) \wedge (l_* \notin Middle) \implies \beta[h_*] > \frac{\epsilon}{2}. \tag{6}$$

$$\neg TERM \wedge (h_* \notin Middle) \wedge (l_* \in Middle) \implies \beta[l_*] > \frac{\epsilon}{2}. \tag{7}$$

$$\neg TERM \wedge (h_* \in Middle) \wedge (l_* \in Middle) \implies \left(\beta[h_*] > \frac{\epsilon}{2}\right) \vee \left(\beta[l_*] > \frac{\epsilon}{2}\right). \tag{8}$$

If neither of h_* and l_* is in $Middle$, then these arms have to be in $Above$ or $Below$. We prove (5) by considering four mutually exclusive cases. Recall that h_* has the lowest lower confidence bound in $High$; l_* has the highest upper confidence bound in Low; and $\hat{p}_{h_*} \geq \hat{p}_{l_*}$.

$$\begin{aligned}
&(\textbf{Case 1}) && (h_* \in Above) \wedge (l_* \in Above) \wedge \neg TERM \\
&\implies && (h_* \in Above) \wedge (l_* \in Above) \\
&\implies && (\forall h \in High : h \in Above) \wedge (l_* \in Above) \\
&\implies && |\{a \in Arms : a \in Above\}| \geq m + 1 \\
&\implies && \exists j \in Bot : j \in Above \\
&\iff && \exists j \in Bot : CROSS_j \implies CROSS. \\
&(\textbf{Case 2}) && (h_* \in Above) \wedge (l_* \in Below) \wedge \neg TERM \\
&\implies && (\hat{p}_{h_*} - \beta[h_*] > c) \wedge (\hat{p}_{l_*} + \beta[l_*] < c) \\
& && \wedge (\hat{p}_{l_*} + \beta[l_*] - \hat{p}_{h_*} + \beta[h_*] > \epsilon) \\
&\implies && (\hat{p}_{l_*} + \beta[l_*] - \hat{p}_{h_*} + \beta[h_*] < 0) \\
& && \wedge (\hat{p}_{l_*} + \beta[l_*] - \hat{p}_{h_*} + \beta[h_*] > \epsilon) \iff \phi. \\
&(\textbf{Case 3}) && (h_* \in Below) \wedge (l_* \in Above) \wedge \neg TERM \\
&\implies && (h_* \in Below) \wedge (l_* \in Above) \\
&\implies && (\hat{p}_{h_*} + \beta[h_*] < c) \wedge (\hat{p}_{l_*} - \beta[l_*] > c) \\
&\implies && \hat{p}_{h_*} < \hat{p}_{l_*} \iff \phi. \\
&(\textbf{Case 4}) && (h_* \in Below) \wedge (l_* \in Below) \wedge \neg TERM \\
&\implies && CROSS. \text{ \{Similar to Case 1.\}}
\end{aligned}$$

Similarly, we show (6) by proving two disjoint cases (for Case 1 we use that $\hat{p}_{h_*} \geq \hat{p}_{l_*}$).

$$\begin{aligned}
&(\textbf{Case 1}) && \neg TERM \wedge (h_* \in Middle) \wedge (l_* \in Above) \\
&\implies && (\hat{p}_{l_*} + \beta[l_*] - \hat{p}_{h_*} + \beta[h_*] > \epsilon) \\
& && \wedge (\hat{p}_{h_*} - \beta[h_*] < c) \wedge (\hat{p}_{l_*} - \beta[l_*] > c) \\
&\implies && (\hat{p}_{h_*} - \beta[h_*] < c) \wedge (\hat{p}_{h_*} + \beta[h_*] > c + \epsilon) \\
&\implies && \beta[h_*] > \frac{\epsilon}{2}.
\end{aligned}$$

$$\begin{aligned}
&(\textbf{Case 2}) && \neg TERM \wedge (h_* \in Middle) \wedge (l_* \in Below) \\
&\implies && (\hat{p}_{l_*} + \beta[l_*] - \hat{p}_{h_*} + \beta[h_*] > \epsilon) \\
& && \wedge (\hat{p}_{h_*} + \beta[h_*] > c) \wedge (\hat{p}_{l_*} + \beta[l_*] < c) \\
&\implies && (\hat{p}_{h_*} + \beta[h_*] > c) \wedge (\hat{p}_{h_*} - \beta[h_*] < c - \epsilon) \\
&\implies && \beta[h_*] > \frac{\epsilon}{2}.
\end{aligned}$$

The proof for (7) is similar. To complete the proof of the lemma, we prove (8):

$$\begin{aligned}
& && \neg TERM \wedge (h_* \in Middle) \wedge (l_* \in Middle) \\
&\implies && \neg TERM \implies \hat{p}_{l_*} + \beta[l_*] - \hat{p}_{h_*} + \beta[h_*] > \epsilon \\
&\implies && \beta[h_*] + \beta[l_*] > \epsilon. \\
&\implies && \left(\beta[h_*] > \frac{\epsilon}{2}\right) \vee \left(\beta[l_*] > \frac{\epsilon}{2}\right). \quad \square
\end{aligned}$$

The above lemma shows that if no arm has crossed and no arm is needy, then the LUCB algorithm must stop. Next we consider a specific confidence bound, β_1, for which we bound the probability of arms crossing or staying needy for long.

3.3. LUCB1

We define Algorithm LUCB1 to be an instance of LUCB that uses the stopping rule from Section 3.1 and the greedy sampling strategy in Section 3.2, while using the following confidence bound:

$$\beta_1(u, t) \stackrel{\text{def}}{=} \sqrt{\frac{1}{2u} \ln\left(\frac{k_1 n t^4}{\delta}\right)}, \text{ where } k_1 = \frac{5}{4}.$$

Note that β_1 satisfies the requirement on β in (4). In the remainder of this section, we bound the expected sample complexity of LUCB1 (Lemma 5 implies that with probability 1, LUCB1 terminates in a finite number of rounds, and thus, is (ϵ, m, δ)-optimal).

Lemma 3. *Under* LUCB1*:* $\mathbb{P}\{CROSS^t\} \leq \frac{\delta}{k_1 t^3}$.

Proof. The proof follows directly from the definition of $CROSS^t$, and with applications of Hoeffding's inequality and the union bound (over arms, and over the possible number of pulls for each arm). $\square$

For sufficiently large t, we define $u_1^*(a, t)$ as an adequate number of samples of arm a such that $\beta_1(u_1^*(a, t), t)$ is no greater than $[\Delta_a \vee \frac{\epsilon}{2}]$:

$$u_1^*(a, t) \stackrel{\text{def}}{=} \left\lceil \frac{1}{2\left[\Delta_a \vee \frac{\epsilon}{2}\right]^2} \ln\left(\frac{k_1 n t^4}{\delta}\right) \right\rceil.$$

The following lemma then shows that the probability that arm a remains needy despite being sampled for more than $4u_1^*(a, t)$ rounds is small.

Lemma 4. *Under* LUCB1*:*

$$\mathbb{P}\{\exists a \in Arms : \left(u_a^t > 4u_1^*(a, t)\right) \wedge NEEDY_a^t\} \leq \frac{3\delta H^{\epsilon/2}}{4 k_1 n t^4}.$$

Proof. Consider an arm a in $Arms$. If $\Delta_a \leq \frac{\epsilon}{2}$, we obtain for $u_a^t > 4u_1^*(a, t)$ that $\beta_1(u_a^t, t) < \frac{\epsilon}{4}$, which implies $\neg NEEDY_a^t$. Now, let us consider the case that $\Delta_a > \frac{\epsilon}{2}$, which is less trivial. Without loss of generality, we may assume that $a \in Top$. Then, by substituting for β_1, and using (3), we get:

$$\begin{aligned}
& \mathbb{P}\{\left(u_a^t > 4u_1^*(a, t)\right) \wedge NEEDY_a^t\} \\
\leq\ & \mathbb{P}\left\{\left(u_a^t > 4u_1^*(a, t)\right) \wedge \left(a \in Middle^t\right)\right\} \\
\leq\ & \mathbb{P}\left\{\left(u_a^t > 4u_1^*(a, t)\right) \wedge \left(\hat{p}_a^t - \beta_1\left(u_a^t, t\right) < c\right)\right\} \\
\leq\ & \sum_{u = 4u_1^*(a,t)+1}^{\infty} \exp\left(-2u\left(p_a - c - \beta_1\left(u_a^t, t\right)\right)^2\right) \\
\leq\ & \sum_{u = 4u_1^*(a,t)+1}^{\infty} \exp\left(-2u\left(\frac{\Delta_a}{2} - \sqrt{\frac{1}{2u}\ln\left(\frac{k_1 n t^4}{\delta}\right)}\right)^2\right) \\
\leq\ & \sum_{u = 4u_1^*(a,t)+1}^{\infty} \exp\left(-2\Delta_a^2\left(\sqrt{u} - \sqrt{u_1^*(a, t)}\right)^2\right) \\
\leq\ & \frac{3\delta}{4\Delta_a^2 k_1 n t^4}.
\end{aligned} \tag{9}$$

The derivation for the last step is in Kalyanakrishnan's Ph.D. thesis (2011, see Appendix B.2). From (9):

$$\begin{aligned}
& \mathbb{P}\{\exists a \in Arms : \left(u_a^t > 4u_1^*(a, t)\right) \wedge NEEDY_a^t\} \\
\leq\ & \frac{3\delta}{4k_1 n t^4} \sum_{a \in Arms, \Delta_a > \frac{\epsilon}{2}} \frac{1}{\Delta_a^2} \leq \frac{3\delta H^{\epsilon/2}}{4k_1 n t^4}. \quad \square
\end{aligned}$$

We now combine the results of Lemma 3 and Lemma 4 to upper-bound the probability that LUCB does not terminate beyond a certain number of rounds T_1^*.

Lemma 5. *Let* $T_1^* = \left\lceil 146 H^{\epsilon/2} \ln\left(\frac{H^{\epsilon/2}}{\delta}\right)\right\rceil$. *For every* $T \geq T_1^*$, *the probability that* LUCB1 *has not terminated after* T *rounds of sampling is at most* $\frac{4\delta}{T^2}$.

Proof. Let $\overline{T} = \left\lceil \frac{T}{2} \right\rceil$. We define two events, E_1 and E_2, over the interval $\{\overline{T}, \overline{T}+1, \ldots, T-1\}$:

$$E_1 \stackrel{\text{def}}{=} \exists t \in \{\overline{T}, \overline{T}+1, \ldots, T-1\} : CROSS^t, \text{ and}$$

$$\begin{aligned}
E_2 \stackrel{\text{def}}{=}\ & \exists t \in \{\overline{T}, \overline{T}+1, \ldots, T-1\}\ \exists a \in Arms : \\
& \left(u_a^t > 4u_1^*(a, t)\right) \wedge NEEDY_a^t.
\end{aligned}$$

We show that if neither E_1 nor E_2 occurs, then LUCB1 must necessarily terminate after at most T rounds. If the algorithm terminates after some $t \leq \overline{T}$ rounds, there is nothing left to prove. Consider the case that the algorithm has not terminated after $\overline{T}$ rounds, and neither E_1 nor E_2 occurs. In this case, let *#rounds* be the number of additional rounds of sampling, up to round T. Applying Lemma 2, we get:

$$\begin{aligned}
\#rounds &= \sum_{t=\overline{T}}^{T-1} \mathbf{1}\left[NEEDY^t_{h^t_*} \vee NEEDY^t_{l^t_*}\right] \\
&\leq \sum_{t=\overline{T}}^{T-1} \sum_{a \in Arms} \mathbf{1}\left[(a = h^t_* \vee a = l^t_*) \wedge NEEDY^t_a\right] \\
&= \sum_{t=\overline{T}}^{T-1} \sum_{a \in Arms} \mathbf{1}\left[(a = h^t_* \vee a = l^t_*) \wedge \left(u^t_a \leq 4u^*_1(a,t)\right)\right] \\
&\leq \sum_{t=\overline{T}}^{T-1} \sum_{a \in Arms} \mathbf{1}\left[(a = h^t_* \vee a = l^t_*) \wedge \left(u^t_a \leq 4u^*_1(a,T)\right)\right] \\
&= \sum_{a \in Arms} \sum_{t=\overline{T}}^{T-1} \mathbf{1}\left[(a = h^t_* \vee a = l^t_*) \wedge \left(u^t_a \leq 4u^*_1(a,T)\right)\right] \\
&\leq \sum_{a \in Arms} 4u^*_1(a,T).
\end{aligned}$$

It is seen that $T \geq T^*_1 \implies T > 2 + 8\sum_{a \in Arms} u^*_1(a,T)$ (Kalyanakrishnan, 2011, see Appendix B.3). Thus, if neither E_1 nor E_2 occurs, the total number of rounds for which LUCB1 lasts is at most $\overline{T} + \#rounds \leq \left\lceil \frac{T}{2} \right\rceil + \sum_{a \in Arms} 4u^*_1(a,T) < T$. Consequently the probability that LUCB1 has not terminated after T rounds can be upper-bounded by $\mathbb{P}\{E_1 \vee E_2\}$. Applying Lemma 3 and Lemma 4, we obtain:

$$\begin{aligned}
\mathbb{P}\{E_1 \vee E_2\} &\leq \sum_{t=\overline{T}}^{T-1} \left(\frac{\delta}{k_1 t^3} + \frac{3\delta H^{\epsilon/2}}{4k_1 n t^4}\right) \\
&\leq \left(\frac{T}{2}\right)\left(\frac{8\delta}{k_1 T^3}\right)\left(1 + \frac{3H^{\epsilon/2}}{2nT}\right) < \frac{4\delta}{T^2}. \quad \square
\end{aligned}$$

Lemma 5 directly yields a bound on the expected sample complexity, and a related high-probability bound.

Theorem 6. *The expected sample complexity of* LUCB1 *is* $O\left(H^{\epsilon/2} \log\left(\frac{H^{\epsilon/2}}{\delta}\right)\right)$.

Corollary 7. *With probability at least* $1 - \delta$, LUCB1 *terminates after* $O\left(H^{\epsilon/2} \log\left(\frac{H^{\epsilon/2}}{\delta}\right)\right)$ *rounds.*

Proof. From Lemma 5, it follows that the expected sample complexity of LUCB1 is at most $2\left(T^*_1 + \sum_{t=T^*_1+1}^{\infty} \frac{4\delta}{t^2}\right) < 292 H^{\epsilon/2} \ln\left(\frac{H^{\epsilon/2}}{\delta}\right) + 16$. The corollary follows trivially from Lemma 5. $\square$

To the best of our knowledge, the expected sample complexity bound in Theorem 6 is novel even for the $m = 1$ case. For Explore-1, Even-Dar et al. (2006, see Remark 9) do provide a high-probability bound that essentially matches our bound in Corollary 7. However, their elimination algorithm could incur high sample complexity on the δ-fraction of the runs on which mistakes are made—we think it unlikely that elimination algorithms can yield an expected sample complexity bound smaller than $\Omega\left(H^{\epsilon/2} \log\left(\frac{n}{\epsilon\delta}\right)\right)$.

4. Worst Case Sample Complexity Lower Bound

We note that it is easy to restrict the worst case (and therefore, the expected) sample complexity of LUCB1 to $O\left(\frac{n}{\epsilon^2} \log\left(\frac{m}{\delta}\right)\right)$ by running it with $\delta' = \frac{\delta}{2}$, and if it does not terminate after $O\left(\frac{n}{\epsilon^2} \log\left(\frac{m}{\delta}\right)\right)$ rounds, restarting and running Halving instead (again with $\delta' = \frac{\delta}{2}$). Under such a scheme, the mistake probability is at most δ; the expected sample complexity is within a constant factor of LUCB's; and the worst case sample complexity within a constant factor of Halving's.

The remainder of this section is devoted to proving that indeed the worst case sample complexity of Halving is optimal up to a constant factor. Section 4.2.1 in our proof is based on a similar proof provided by Mannor & Tsitsiklis (2004, see Section 3) for $m = 1$. In Section 4.2.2 we present a novel way to aggregate error terms from different bandit instances, which is key for getting an $\Omega(\log(m))$ dependence in our bound.

Theorem 8. *For* $0 < \epsilon \leq \sqrt{\frac{1}{32}}$, $0 < \delta \leq \frac{1}{4}$, $m \geq 6$, $n \geq 2m$, *and every* (ϵ, m, δ)*-optimal algorithm* $\mathcal{A}$, *there is a bandit instance on which* $\mathcal{A}$ *has a worst case sample complexity of at least* $\frac{1}{18375} \cdot \frac{n}{\epsilon^2} \ln\left(\frac{m}{8\delta}\right)$.

To prove the theorem, we consider two sets of bandit instances ($\mathcal{I}_{m-1}$ and $\mathcal{I}_m$), and an (ϵ, m, δ)-optimal algorithm $\mathcal{A}$. We show that if $\mathcal{A}$ has a low worst case sample complexity on instances in $\mathcal{I}_{m-1}$, it must have a high mistake probability on some instance in $\mathcal{I}_m$, thereby contradicting that it is (ϵ, m, δ)-optimal. In this section, we drop the convention that arms are indexed in non-increasing order of their means. Also, we index the arms $0, 1, \ldots, n-1$.

4.1. Bandit Instances

For $l = m-1, m$, let $\mathcal{I}_l$ be the set of all l-sized subsets of arms, excluding arm 0; that is:

$$\mathcal{I}_l \stackrel{\text{def}}{=} \{I \subseteq \{1, 2, \ldots, n-1\} : |I| = l\}.$$

For $I \subseteq \{1, 2, \ldots, n-1\}$ we define by

$$\bar{I} \stackrel{\text{def}}{=} \{1, 2, \ldots, n-1\} \setminus I$$

the set of remaining arms, again excluding arm 0. For each $I \in \mathcal{I}_{m-1} \cup \mathcal{I}_m$, we associate an n-armed bandit instance $\mathcal{B}_I$, in which each arm a yields rewards from a Bernoulli distribution with mean as follows:

$$p_a = \begin{cases} 1/2 & \text{if } a = 0, \\ 1/2 + 2\epsilon & \text{if } a \in I, \\ 1/2 - 2\epsilon & \text{if } a \in \bar{I}. \end{cases}$$

Observe that every instance in $\mathcal{I}_{m-1}$ and $\mathcal{I}_m$ has exactly m (ϵ, m)-optimal arms. In the former case, arm 0 is among these (ϵ, m)-optimal arms, while in the latter, it is not. We build on the intuition that it is difficult for an algorithm to recognize this distinction without sampling the arms a sufficiently large number of times.

4.2. Bound

To derive our bound, we make the following assumption, and then proceed to establish a contradiction.

Assumption 9. *Assume there is an (ϵ, m, δ)-optimal algorithm $\mathcal{A}$ that for each bandit problem $\mathcal{B}_I$, $I \in \mathcal{I}_{m-1}$, has sample complexity of at most $C\frac{n}{\epsilon^2}\ln\left(\frac{m}{8\delta}\right)$, where $C = \frac{1}{18375}$.*

We denote by $\mathbb{P}_I$ the probability distribution that results from the bandit problem $\mathcal{B}_I$ and possible randomization of the sampling algorithm $\mathcal{A}$. Also we denote by $S_{\mathcal{A}}$ the set of arms that $\mathcal{A}$ returns, and by T_i the number of times algorithm $\mathcal{A}$ samples arm i before terminating. Then, for all $I \in \mathcal{I}_{m-1}$,

$$\mathbb{P}_I\left\{S_{\mathcal{A}} = I \cup \{0\}\right\} \geq 1 - \delta, \tag{10}$$

since $\mathcal{A}$ is (ϵ, m, δ)-optimal. From the bound on the sample complexity, we have for all $I \in \mathcal{I}_{m-1}$ that

$$\mathbb{E}_I\left[\sum_{i=0}^{n-1} T_i\right] \leq C\frac{n}{\epsilon^2}\ln\left(\frac{m}{8\delta}\right). \tag{11}$$

4.2.1. Changing $\mathbb{P}_I$ to $\mathbb{P}_{I\cup\{j\}}$

Consider a fixed $I \in \mathcal{I}_{m-1}$. From (11) we get that there are at most $\frac{n}{4}$ arms $j \in \bar{I}$ with $\mathbb{E}_I[T_j] \geq C\frac{4}{\epsilon^2}\ln\left(\frac{m}{8\delta}\right)$. Thus, there are (since $n \geq 2m$) at least $n - m - \frac{n}{4} \geq \frac{(n-m)}{2}$ arms $j \in \bar{I}$ with $\mathbb{E}_I[T_j] < \frac{4C}{\epsilon^2}\ln\left(\frac{m}{8\delta}\right)$. For these arms, Markov's inequality gives

$$\mathbb{P}_I\left\{T_j \geq \frac{16C}{\epsilon^2}\ln\left(\frac{m}{8\delta}\right)\right\} < \frac{1}{4}. \tag{12}$$

Let $T^* \stackrel{\text{def}}{=} \frac{16C}{\epsilon^2}\ln\left(\frac{m}{8\delta}\right)$ and $\Delta \stackrel{\text{def}}{=} 2\epsilon T^* + \sqrt{T^*}$. Also, let K_j denote the sum of rewards received for arm j.

Lemma 10. *Let $I \in \mathcal{I}_{m-1}$ and $j \in \bar{I}$. Then,*

$$\mathbb{P}_I\left\{T_j \leq T^*, K_j \leq \frac{T_j}{2} - \Delta\right\} \leq \frac{1}{4}.$$

Proof. Let $K_j(t)$ denote the sum of rewards of arm j after t trials of arm j. Kolmogorov's inequality gives $\mathbb{P}_I\left\{\min_{1\leq t\leq T^*}\left(K_j(t) - t\left(\frac{1}{2} - 2\epsilon\right)\right) \leq -\sqrt{T^*}\right\} \leq \frac{1}{4}$. Thus:

$$\begin{aligned}
& \mathbb{P}_I\left\{T_j \leq T^*, K_j \leq \frac{T_j}{2} - \Delta\right\} \\
\leq \quad & \mathbb{P}_I\left\{\min_{1\leq t\leq T^*}\left(K_j(t) - \frac{t}{2}\right) \leq -2\epsilon T^* - \sqrt{T^*}\right\} \leq \frac{1}{4}. \quad \square
\end{aligned}$$

Lemma 11. *Let $I \in \mathcal{I}_{m-1}$ and $j \in \bar{I}$. Let W be some fixed sequence of outcomes (or "run") of algorithm $\mathcal{A}$ with $T_j \leq T^*$ and $K_j \geq \frac{T_j}{2} - \Delta$. Then:*

$$\mathbb{P}_{I\cup\{j\}}\{W\} > \mathbb{P}_I\{W\} \cdot \exp(-32\epsilon\Delta).$$

Proof. We need to bound the decrease in the probability of observing the outcome sequence W when the mean reward of arm j is changed. Since the mean reward of arm j is higher in the bandit problem $\mathcal{B}_{I\cup\{j\}}$, the probability of W is decreased the most when in W, only few 1-rewards are received for arm j. Thus,

$$\begin{aligned}
& \mathbb{P}_{I\cup\{j\}}\{W\} \\
\geq \quad & \mathbb{P}_I\{W\}\frac{\left(\frac{1}{2}+2\epsilon\right)^{\left(\frac{T_j}{2}-\Delta\right)}\cdot\left(\frac{1}{2}-2\epsilon\right)^{\left(\frac{T_j}{2}+\Delta\right)}}{\left(\frac{1}{2}-2\epsilon\right)^{\left(\frac{T_j}{2}-\Delta\right)}\cdot\left(\frac{1}{2}+2\epsilon\right)^{\left(\frac{T_j}{2}+\Delta\right)}} \\
= \quad & \mathbb{P}_I\{W\}\left(\frac{\frac{1}{2}-2\epsilon}{\frac{1}{2}+2\epsilon}\right)^{2\Delta} > \mathbb{P}_I\{W\}\cdot\exp(-32\epsilon\Delta)
\end{aligned}$$

since $\left(\frac{\frac{1}{2}-2\epsilon}{\frac{1}{2}+2\epsilon}\right)^{2\Delta} > \exp(-32\epsilon\Delta)$ for $0 < \epsilon \leq \frac{1}{\sqrt{32}}$. $\square$

Lemma 12. *If (12) holds for $I \in \mathcal{I}_{m-1}$ and $j \in \bar{I}$, then for any set $\mathcal{W}$ of sequences of outcomes W,*

$$\mathbb{P}_{I\cup\{j\}}\{\mathcal{W}\} > \left(\mathbb{P}_I\{\mathcal{W}\} - \frac{1}{2}\right)\cdot\frac{8\delta}{m}.$$

In particular,

$$\mathbb{P}_{I\cup\{j\}}\left\{S_{\mathcal{A}} = I \cup \{0\}\right\} > \frac{2\delta}{m}. \tag{13}$$

Proof. In this proof, we use the explicit notation T_j^W and K_j^W to denote the number of trials and number of 1-rewards, respectively, of arm j for some fixed sequence of outcomes W. By applying Lemma 11, Lemma 10, and (12) in sequence, we get:

$$\begin{aligned}
& \mathbb{P}_{I\cup\{j\}}\{\mathcal{W}\} = \mathbb{P}_{I\cup\{j\}}\{W : W \in \mathcal{W}\} \\
\geq \quad & \mathbb{P}_{I\cup\{j\}}\left\{W : W \in \mathcal{W}, T_j^W \leq T^*, K_j^W \geq T_j^W - \Delta\right\} \\
\geq \quad & \mathbb{P}_I\left\{W : W \in \mathcal{W}, T_j^W \leq T^*, K_j^W \geq T_j^W - \Delta\right\}\cdot\exp(-32\epsilon\Delta) \\
\geq \quad & \left(\mathbb{P}_I\left\{W : W \in \mathcal{W}, T_j^W \leq T^*\right\} - \frac{1}{4}\right)\cdot\exp(-32\epsilon\Delta) \\
\geq \quad & \left(\mathbb{P}_I\{W : W \in \mathcal{W}\} - \frac{1}{2}\right)\cdot\exp(-32\epsilon\Delta) \\
> \quad & \left(\mathbb{P}_I\{\mathcal{W}\} - \frac{1}{2}\right)\cdot\frac{8\delta}{m}.
\end{aligned}$$

The last step follows from the observation that for $\delta \leq \frac{1}{4}$, $m \geq 6$, and $C = \frac{1}{18375}$, we have $32\epsilon\Delta < \ln\left(\frac{m}{8\delta}\right)$. To obtain (13), observe from (10) that $\mathbb{P}_I\left\{S_{\mathcal{A}} = I \cup \{0\}\right\} \geq 1 - \delta \geq \frac{3}{4}$. $\square$

4.2.2. Summing over $\mathcal{I}_{m-1}$ and $\mathcal{I}_m$

Now we present the main innovation in our proof for generalizing the lower bound from Explore-1 to Explore-m. We carefully aggregate error terms over bandit instances in $\mathcal{I}_{m-1}$ and $\mathcal{I}_m$ to show that if Assumption 9 is true, then some bandit instance in $\mathcal{I}_m$ must exhibit a mistake probability in excess of δ. First we obtain the following inequality.

$$\begin{aligned}
&\sum_{J \in \mathcal{I}_m} \mathbb{P}_J \{S_\mathcal{A} \neq J\} \\
\geq\ & \sum_{J \in \mathcal{I}_m} \sum_{j \in J} \mathbb{P}_J \{S_\mathcal{A} = \{J \cup \{0\}\} \setminus \{j\}\} \\
=\ & \sum_{J \in \mathcal{I}_m} \sum_{j \in J} \sum_{I \in \mathcal{I}_{m-1}} 1[I \cup \{j\} = J] \cdot \mathbb{P}_J\{S_\mathcal{A} = I \cup \{0\}\} \\
=\ & \sum_{J \in \mathcal{I}_m} \sum_{j=1}^{n-1} \sum_{I \in \mathcal{I}_{m-1}} 1[I \cup \{j\} = J] \cdot \mathbb{P}_J\{S_\mathcal{A} = I \cup \{0\}\} \\
=\ & \sum_{I \in \mathcal{I}_{m-1}} \sum_{J \in \mathcal{I}_m} \sum_{j=1}^{n-1} 1[I \cup \{j\} = J] \cdot \mathbb{P}_J\{S_\mathcal{A} = I \cup \{0\}\} \\
=\ & \sum_{I \in \mathcal{I}_{m-1}} \sum_{J \in \mathcal{I}_m} \sum_{j \in \bar{I}} 1[I \cup \{j\} = J] \cdot \mathbb{P}_J\{S_\mathcal{A} = I \cup \{0\}\} \\
=\ & \sum_{I \in \mathcal{I}_{m-1}} \sum_{j \in \bar{I}} \sum_{J \in \mathcal{I}_m} 1[I \cup \{j\} = J] \cdot \mathbb{P}_J\{S_\mathcal{A} = I \cup \{0\}\} \\
=\ & \sum_{I \in \mathcal{I}_{m-1}} \sum_{j \in \bar{I}} \mathbb{P}_{I \cup \{j\}}\{S_\mathcal{A} = I \cup \{0\}\}.
\end{aligned}$$

Next we note that for $I \in \mathcal{I}_{m-1}$, (13) holds for at least $\frac{n-m}{2}$ arms $j \in \bar{I}$. Summing the corresponding errors for these arms yields

$$\sum_{J \in \mathcal{I}_m} \mathbb{P}_J \{S_\mathcal{A} \neq J\} > \sum_{I \in \mathcal{I}_{m-1}} \frac{n-m}{2} \cdot \frac{2\delta}{m}$$
$$= \binom{n-1}{m-1} \frac{n-m}{m} \delta = \binom{n-1}{m} \delta = |\mathcal{I}_m| \delta.$$

Hence, in contradiction to Assumption 9, there exists a bandit instance $J \in \mathcal{I}_m$ with $\mathbb{P}_J \{S_\mathcal{A} \neq J\} > \delta$.

5. Conclusion

We have addressed the problem of subset selection, which generalizes the problem of single-arm selection. Under a PAC setting, we have presented the LUCB algorithm and provided a novel expected sample complexity bound for subset (and single-arm) selection. We have also given a worst case sample complexity lower bound for subset selection.

It is interesting to note the similarity between the LUCB and UCB (Auer et al., 2002) algorithms. Sampling greedily with respect to suitable lower and upper confidence bounds, LUCB achieves the best-known PAC bounds; being greedy with respect to upper confidence bounds alone, UCB yields minimal cumulative regret. It would be worthwhile to investigate if there is a deeper connection between these two algorithms and their respective settings.

We conjecture that the expected sample complexity of (ϵ, m, δ)-optimal algorithms for Explore-m is at least $\Omega\left(H^{\epsilon/2} \log\left(\frac{1}{\delta}\right)\right)$, which would indicate that $H^{\epsilon/2}$ is indeed an appropriate measure of problem complexity. However, at present we are unaware of a way to improve the $O\left(H^{\epsilon/2} \log\left(\frac{H^{\epsilon/2}}{\delta}\right)\right)$ upper bound achieved by LUCB1, unless $H^{\epsilon/2}$ is provided as input to the algorithm. On a closely-related exploration problem, Audibert et al. (2010) also observe that a tighter upper bound can be achieved if the problem complexity (a quantity similar to $H^{\epsilon/2}$) is known beforehand.

Acknowledgments

A major portion of Shivaram Kalyanakrishnan's contribution to this paper was made when he was a graduate student in the Department of Computer Science, The University of Texas at Austin. The authors are grateful to anonymous reviewers for their comments. Peter Stone is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and FHWA (DTFH61-07-H-00030).

References

Audibert, Jean-Yves, Bubeck, Sébastien, and Munos, Rémi. Best arm identification in multi-armed bandits. In *Proc. COLT 2010*, pp. 41–53. Omnipress, 2010.

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

Even-Dar, Eyal, Mannor, Shie, and Mansour, Yishay. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *JMLR*, 7:1079–1105, 2006.

Kalyanakrishnan, Shivaram. *Learning Methods for Sequential Decision Making with Imperfect Representations.* PhD thesis, Department of Computer Science, The University of Texas at Austin, December 2011.

Kalyanakrishnan, Shivaram and Stone, Peter. Efficient selection of multiple bandit arms: Theory and practice. In *Proc. ICML 2010*, pp. 511–518. Omnipress, 2010.

Mannor, Shie and Tsitsiklis, John N. The sample complexity of exploration in the multi-armed bandit problem. *JMLR*, 5:623–648, 2004.

Mnih, Volodymyr, Szepesvári, Csaba, and Audibert, Jean-Yves. Empirical Bernstein stopping. In *Proc. ICML 2008*, pp. 672–679. ACM, 2008.

Schuurmans, Dale and Greiner, Russell. Sequential PAC learning. In *Proc. COLT 1995*, pp. 377–384. ACM, 1995.