In Fuernkranz, Joachims, editors, *Proceedings of the Twenty-seventh International Conference on Machine Learning*, pp. 511--518, Omnipress, 2010.

Efficient Selection of Multiple Bandit Arms: Theory and Practice

Shivaram Kalyanakrishnan SHIVARAM@CS.UTEXAS.EDU
Peter Stone PSTONE@CS.UTEXAS.EDU
Department of Computer Science, The University of Texas at Austin, Austin Texas 78712-0233, USA

Abstract

We consider the general, widely applicable problem of selecting from n real-valued random variables a subset of size m of those with the highest means, based on as few samples as possible. This problem, which we denote Explore-m, is a core aspect in several stochastic optimization algorithms, and applications of simulation and industrial engineering. The theoretical basis for our work is an extension of a previous formulation using multi-armed bandits that is devoted to identifying just the one best of n random variables (Explore-1). In addition to providing PAC bounds for the general case, we tailor our theoretically grounded approach to work efficiently in practice. Empirical comparisons of the resulting sampling algorithm against state-of-the-art subset selection strategies demonstrate significant gains in sample efficiency.

1. Introduction

We consider the problem of *efficient subset selection*: given n real-valued random variables, our task is to reliably identify m among them with the highest means, while keeping the total number of samples minimal. This problem merits attention from several fields, including industrial engineering (Koenig & Law, 1985), simulation (Kim & Nelson, 2001), and evolutionary computation (Schmidt et al., 2006).

The theoretical basis for our work is an extension of recent research from Even-Dar et al. (2006) (and Mannor & Tsitsiklis (2004)), who consider a problem of "pure exploration" in a multi-armed bandit (Berry & Fristedt, 1985). Unlike traditional bandit problems in which the rewards accrued *during* the exploratory phase must be maximized (and thus the regret minimized), in their problem the aim is to minimize the total number of samples needed to reliably identify an arm with an expected reward that is within ϵ of the maximum achievable (an "ϵ-optimal" arm). The parameter ϵ serves to specify tolerance. Casting the problem into a PAC framework, Even-Dar et al. (2006) provide an algorithm that identifies an ϵ-optimal arm in an n-armed bandit with probability at least $1 - \delta$, incurring a sample complexity that is $O(\frac{n}{\epsilon^2} log(\frac{1}{\delta}))$. This sample complexity *matches*, up to constants, a lower bound derived by Mannor & Tsitsiklis (2004).

In this paper we generalize the problem formulated by Even-Dar et al. (2006): our aim is to identify m arms of an n-armed bandit such that the expected reward of each chosen arm is at least $p_m - \epsilon$, where p_m is the m^{th} highest expected reward among the bandit's arms. We denote this problem "Explore-m" (thus, the special case studied by Even-Dar et al. (2006) is Explore-1). We extend the "median elimination" algorithm proposed by Even-Dar et al. (2006) to provide an algorithm for Explore-m that satisfies a PAC constraint similar to the one for Explore-1, while achieving a sample complexity that is $O(\frac{n}{\epsilon^2} log(\frac{m}{\delta}))$. We argue that in practice, sample efficiency can be further improved by adapting to data through a sequential procedure, while preserving the PAC guarantee. Empirical results affirm significant gains for the resulting adaptive method when compared with state-of-the-art subset selection methods (Chen et al., 2008; Heidrich-Meisner & Igel, 2009).

This paper is organized as follows. We define the Explore-m problem in Section 2. In Section 3 we provide three PAC algorithms for Explore-m and analyze their sample complexity. In sections 2 and 3 we closely follow the presentation style of Even-Dar et al. (2006). In Section 4 we present an adaptive method for solving Explore-m in practice. Empirical evaluation and comparisons follow in Section 5. We discuss related work and conclude in Section 6.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

2. Problem Statement

We consider an n-armed bandit with its arms numbered $1, 2, \ldots, n$. Each sample (or "pull") of arm a yields a reward of either 0 or 1, generated randomly from a Bernoulli distribution with mean p_a. Without loss of generality we assume that

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

The arm distributions are independent, and do not depend on the history of pulls.[1] Arm a is defined to be (ϵ, m)-optimal, $\forall \epsilon > 0$, $\forall m \in \{1, 2, \ldots, n\}$, if

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

To solve the Explore-m problem, an algorithm may sample arms of the n-armed bandit and record the results of its pulls; the algorithm is required to terminate and return a set of m arms. Such an algorithm $\mathcal{L}$ is defined to be (ϵ, m, δ)-optimal, $\forall \delta \in (0, 1)$, if with probability at least $1 - \delta$, *each* of the arms it returns is (ϵ, m)-optimal. Note that we do not require the m arms returned by $\mathcal{L}$ to be in any particular order. The *sample complexity* of $\mathcal{L}$ is the total number of pulls it performs before termination.

Let us denote by T_i the set $\{1, 2, \ldots, i\}$. From (1) and (2) we see that every arm in T_m is (ϵ, m)-optimal. Hence, there are *at least* m (ϵ, m)-optimal arms, and so the Explore-m problem is well-defined. Let us denote by B the set of arms that are *not* (ϵ, m)-optimal. In general $0 \leq |B| \leq (n - m)$. Note that Explore-m is a trivial problem when $m = n$.

3. Algorithms and PAC Analysis

Even-Dar et al. (2006) present two $(\epsilon, 1, \delta)$-optimal algorithms for Explore-1: a naïve algorithm that achieves a sample complexity $O(\frac{n}{\epsilon^2} log(\frac{n}{\delta}))$, and a "median elimination" algorithm that improves the sample complexity to $O(\frac{n}{\epsilon^2} log(\frac{1}{\delta}))$. We generalize these existing algorithms to construct three (ϵ, m, δ)-optimal algorithms for Explore-m. Our first algorithm, "Direct," essentially implements the naïve strategy of pulling each arm a fixed number of times. The second algorithm, "Incremental," uses the median elimination algorithm as a subroutine. The sample complexities of both methods are improved by our third algorithm, "Halving," which modifies the median elimination algorithm to identify m arms instead of 1. In the following text, ln denotes the natural logarithm and log denotes the logarithm to the base 2.

DIRECT: Under Direct (Algorithm 1), arms are sampled a fixed number of times (line 2) such that with high probability the m arms with the highest *empirical* averages (denoted $\hat{p}$) are all (ϵ, m)-optimal.

Algorithm 1 Direct(n, m, ϵ, δ)

```
1: for all $a \in T_n$ do
2:     Sample arm $a$ $\lceil \frac{2}{\epsilon^2} ln(\frac{n}{\delta}) \rceil$ times; let $\hat{p}_a$ be its average reward.
3: end for
4: Find $S \subset T_n$ such that $|S| = m$, and $\forall i \in S$ $\forall j \in (T_n - S)$: $(\hat{p}_i \geq \hat{p}_j)$.
5: Return $S$.
```

Theorem 1. Direct(n, m, ϵ, δ) *is* (ϵ, m, δ)*-optimal with sample complexity* $O(\frac{n}{\epsilon^2} log(\frac{n}{\delta}))$.

Proof. Recall that B is the set of arms in T_n that are not (ϵ, m)-optimal. From (1) and (2) we can relate T_m and B as follows:

$$\forall i \in T_m \ \forall j \in B : (p_i - p_j > \epsilon). \tag{3}$$

Since $|S| = m$, an arm j in B can occur in S only if there is some arm i in T_m such that $\hat{p}_i \leq \hat{p}_j$ (line 4). In turn (3) implies that the latter event can only occur if $\hat{p}_i \leq p_i - \frac{\epsilon}{2}$ or $\hat{p}_j \geq p_j + \frac{\epsilon}{2}$. Switching to probabilities, applying the union bound and Hoeffding's inequality, we get:

$$\begin{aligned}
&P(\exists j \in B : (j \in S)) \\
&\leq P(\exists i \in T_m : (\hat{p}_i \leq p_i - \frac{\epsilon}{2})) + P(\exists j \in B : (\hat{p}_j \geq p_j + \frac{\epsilon}{2})) \\
&\leq \sum_{i \in T_m} P(\hat{p}_i \leq p_i - \frac{\epsilon}{2}) + \sum_{j \in B} P(\hat{p}_j \geq p_j + \frac{\epsilon}{2}) \\
&\leq |T_m| e^{-\frac{\epsilon^2}{2} \lceil \frac{2}{\epsilon^2} ln(\frac{n}{\delta}) \rceil} + |B| e^{-\frac{\epsilon^2}{2} \lceil \frac{2}{\epsilon^2} ln(\frac{n}{\delta}) \rceil} \\
&\leq (|T_m| + |B|)(\frac{\delta}{n}) \leq \delta.
\end{aligned}$$

Since each arm is pulled exactly $\lceil \frac{2}{\epsilon^2} ln(\frac{n}{\delta}) \rceil$ times, the sample complexity of Direct is $O(\frac{n}{\epsilon^2} log(\frac{n}{\delta}))$. □

INCREMENTAL: Unlike Direct, Incremental (Algorithm 2) proceeds through m rounds. At the beginning of round l, S_l is the set of arms that have been selected, and R_l the set of arms remaining (line 1). During round l, an $(\epsilon, 1)$-optimal arm in R_l is selected with high probability by invoking the median elimination algorithm (Even-Dar et al., 2006) (line 3). We show that an $(\epsilon, 1)$-optimal arm in R_l is necessarily (ϵ, m)-optimal in T_n.

Algorithm 2 Incremental(n, m, ϵ, δ)

```
1: $S_1 \leftarrow \{\}$; $R_1 \leftarrow T_n$.
2: for $l = 1$ to $m$ do
3:     $a' \leftarrow$Median-Elimination$(R_l, \epsilon, \frac{\delta}{m})$.
4:     $S_{l+1} \leftarrow S_l \cup \{a'\}$; $R_{l+1} \leftarrow R_l - \{a'\}$.
5: end for
6: Return $S_{m+1}$.
```

[1] The analysis and algorithms in this paper easily extend to the case where arm distributions have known, *bounded* ranges, and also when some have *equal* means.

Theorem 2. INCREMENTAL(n, m, ϵ, δ) *is* (ϵ, m, δ)-*optimal with sample complexity* $O(\frac{mn}{\epsilon^2} log(\frac{m}{\delta}))$.

Proof. Since $|R_l| = n-l+1$, and $l \leq m$, R_l necessarily contains an arm a in T_m. Among the true means of the arms in R_l, let p^* be the highest. It follows from (1) and (2) that for any arm a' that is $(\epsilon, 1)$-optimal with respect to R_l, $p_{a'} \geq p^* - \epsilon \geq p_a - \epsilon \geq p_m - \epsilon$: i.e., a' is (ϵ, m)-optimal with respect to T_n. On round l, since MEDIAN-ELIMINATION (line 3) returns an arm that is *not* $(\epsilon, 1)$-optimal in R_l with probability at most $\frac{\delta}{m}$, the probability that INCREMENTAL selects an arm that is not (ϵ, m)-optimal is at most δ.

The sample complexity of INCREMENTAL is derived from its m calls to MEDIAN-ELIMINATION. Since $|R_l| \leq n$, each call performs at most $O(\frac{n}{\epsilon^2} log(\frac{1}{\frac{\delta}{m}}))$ pulls (see Even-Dar et al., 2006, Lemma 12), giving a total sample complexity that is $O(\frac{mn}{\epsilon^2} log(\frac{m}{\delta}))$. □

HALVING: While INCREMENTAL *selects* an arm every round, HALVING (Algorithm 3) *eliminates* multiple arms every round based on their inferior empirical averages. From R_l, the set of arms remaining at the beginning of round l, half proceed to round $l+1$ (except that m proceed to the last round). Arms are sampled enough times each round (line 5) such that at least m (ϵ, m)-optimal arms are likely to survive elimination. Specifically, round l is associated with parameters ϵ_l and δ_l, and we ensure that with probability at least $1 - \delta_l$ the m^{th} highest true mean in R_{l+1} is not lower than the m^{th} highest true mean in R_l by more than ϵ_l. The sequences (ϵ_l) and (δ_l) (lines 2 and 9) are designed such that HALVING is (ϵ, m, δ)-optimal with sample complexity $O(\frac{n}{\epsilon^2} log(\frac{m}{\delta}))$.

Algorithm 3 HALVING$(n,\ m,\ \epsilon,\ \delta)$

```
 1: $R_1 \leftarrow T_n$.
 2: $\epsilon_1 \leftarrow \frac{\epsilon}{4}$; $\delta_1 \leftarrow \frac{\delta}{2}$.
 3: for $l = 1$ to $\lceil log(\frac{n}{m}) \rceil$ do
 4:    for all $a \in R_l$ do
 5:       Sample arm $a$ $\lceil \frac{2}{\epsilon_l^2} ln(\frac{3m}{\delta_l}) \rceil$ times; let $\hat{p}_a$ be its average reward.
 6:    end for
 7:    Find $R'_l \subset R_l$ such that $|R'_l| = \max(\lceil \frac{|R_l|}{2} \rceil, m)$, and
       $\forall i \in R_l\ \forall j \in (R_l - R'_l)$: $(\hat{p}_i \geq \hat{p}_j)$.
 8:    $R_{l+1} \leftarrow R'_l$.
 9:    $\epsilon_{l+1} \leftarrow \frac{3}{4}\epsilon_l$; $\delta_{l+1} \leftarrow \frac{1}{2}\delta_l$.
10: end for
11: Return $R_{\lceil log(\frac{n}{m}) \rceil + 1}$.
```

Theorem 3. HALVING(n, m, ϵ, δ) *is* (ϵ, m, δ)-*optimal with sample complexity* $O(\frac{n}{\epsilon^2} log(\frac{m}{\delta}))$.

Proof. Let us sort the arms in R_l in decreasing order of their *true* means. Let a_i^l be the i^{th} arm in the sorted list and let p_i^l be its true mean. We say a "mistake" is made in round l if $p_m^l - p_m^{l+1} > \epsilon_l$. Note that (1) $p_m^1 = p_m$, (2) $\sum_{l=1}^{\lceil log(\frac{n}{m}) \rceil} \epsilon_l < \epsilon$, and (3) $\sum_{l=1}^{\lceil log(\frac{n}{m}) \rceil} \delta_l < \delta$. In effect, it suffices to show that the probability of making a mistake in round l is at most δ_l: this would establish that $P(p_m - p_m^{\lceil log(\frac{n}{m}) \rceil + 1} > \epsilon) < \delta$, or in other words, that HALVING is (ϵ, m, δ)-optimal. We show that for a mistake to occur on round l, at least one of two events, E_1 and E_2, must occur; however, $P(E_1) + P(E_2 | \neg E_1) \leq \delta_l$.

Let $A_l = \{a_i^l, i \in 1, 2, \ldots, m\}$: A_l contains the m arms from R_l with the highest true means. E_1 denotes the event $\exists a \in A_l : (\hat{p}_a \leq p_a - \frac{\epsilon_l}{2})$. By applying Hoeffding's inequality and the union bound we obtain:

$$P(E_1) \leq m e^{-\frac{\epsilon_l^2}{2} \lceil \frac{2}{\epsilon_l^2} ln(\frac{3m}{\delta_l}) \rceil} \leq \frac{\delta_l}{3}. \quad (4)$$

Let $B_l = \{j \in R_l, p_j < p_m^l - \epsilon_l\}$: B_l is the set of arms that are not (ϵ_l, m)-optimal in R_l. We call an arm b in B_l "bad" if its empirical average equals or exceeds that of some arm in A_l. If E_1 does not occur, note that b can be bad only if $\hat{p}_b \geq p_b + \frac{\epsilon_l}{2}$; a similar application of Hoeffding's inequality shows that the probability of the latter event is at most $\frac{\delta_l}{3m}$. Let the number of bad arms in B_l be $\#bad$; counting bad arms as Boolean results of $|B_l|$ coin tosses, we obtain that $E[\#bad | \neg E_1]$ is at most $|B_l| \frac{\delta_l}{3m}$.

E_2 denotes the event that $\#bad \geq |R_{l+1}| - m + 1$. A moment's reflection informs us that if E_1 does not occur, a mistake can be made on round l only if E_2 occurs. Markov's inequality establishes that

$$\begin{aligned} P(E_2 | \neg E_1) &= P(\#bad \geq (|R_{l+1}| - m + 1) | \neg E_1) \\ &\leq \frac{E[\#bad | \neg E_1]}{|R_{l+1}| - m + 1} \leq \frac{|B_l|}{|R_{l+1}| - m + 1} (\frac{\delta_l}{3m}) \\ &\leq \frac{|R_l| - m}{|R_{l+1}| - m + 1} (\frac{\delta_l}{3m}) \leq \frac{2}{3} \delta_l. \quad (5) \end{aligned}$$

Arithmetic for the last step follows from the observation that $|R_l| \leq 2|R_{l+1}|$ (lines 7 and 8). Together, (4) and (5) complete our proof.

In Algorithm 3 the total number of pulls across all rounds (line 5) is $\sum_{l=1}^{\lceil log(\frac{n}{m}) \rceil} \lceil \frac{2|R_l|}{\epsilon_l^2} ln(\frac{3m}{\delta_l}) \rceil$. Slight modifications to the steps for bounding a similar sum when $m = 1$ (see Even-Dar et al., 2006, Lemma 12) establish that it is $O(\frac{n}{\epsilon^2} log(\frac{m}{\delta}))$. □

Indeed HALVING achieves the lowest sample complexity bound $(O(\frac{n}{\epsilon^2} log(\frac{m}{\delta})))$ among the algorithms presented in this section. At present we are unaware of a tighter *lower* bound for EXPLORE-m than the one automatically carrying over from EXPLORE-1 $(\Omega(\frac{n}{\epsilon^2} log(\frac{1}{\delta})))$ (Mannor & Tsitsiklis, 2004). Thus, proving matching upper and lower bounds for EXPLORE-m is an open problem.

4. Adaptive Bounds in Practice

Which instances of Explore-m are easy and which ones are difficult? It seems intuitive that when the top m and bottom $n-m$ arms are separated by a relatively large margin, or when arm distributions have low variances, fewer samples would be needed to reliably identify m (ϵ, m)-optimal arms. However, for given n, m, ϵ, and δ, note that any of the algorithms presented in Section 3 performs the same number of pulls, regardless of the arm distributions. These algorithms are designed to be *sufficient* for achieving a PAC guarantee in the *worst* case, when differences between the arms' true means are small.

In this section we focus on improving sample efficiency in practice by adapting to the spacing between arm means and their variances, guided by their empirical returns. Starting with a conceptually simple "Uniform" algorithm that samples arms equally often, we consider progressive improvements to conserve samples in practice *while retaining the PAC guarantee*. Experiments in Section 5 show that the sequential procedure thus developed achieves significant gains in sample complexity over existing sampling methods.

Recall that numbers $1, 2, \ldots, n$ index the *true* means of the arms in decreasing order. In the absence of knowledge about these indices, let us index the arms $(1), (2), \ldots, (n)$ after each pull such that their *empirical* means are in non-decreasing order, i.e., $\hat{p}_{(1)} \geq \hat{p}_{(2)} \geq \ldots \geq \hat{p}_{(n)}$. Let us separate the highest m arms into a set $High = \{(1), (2), \ldots, (m)\}$ and leave the rest in the set $Low = \{(m+1), (m+2), \ldots, (n)\}$.

With each arm h in $High$ let us associate numbers δ_h and LB_h such that by Hoeffding's inequality, with probability at least $1 - \delta_h$, $p_h \geq LB_h$. Likewise for each arm l in Low we ensure that with probability at least $1 - \delta_l$, $UB_l \geq p_l$. Here $LB_{(i)}$ and $UB_{(i)}$ are respectively lower and upper bounds on the true mean of (i). If we return $High$ as our answer, we can meet PAC requirements by ensuring that (a) $\sum_{i=1}^{n} \delta_{(i)} \leq \delta$, and (b) $\forall h \in High, \forall l \in Low : (LB_h + \epsilon \geq UB_l)$.

If we adopt the convention of adding ϵ to the sample means (and lower bounds) of arms in $High$, criterion (b) essentially states that lower bounds of arms in $High$ and upper bounds of arms in Low must not collide [2]. The Direct algorithm in Section 3 precisely achieves criteria (a) and (b) by setting $\delta_i = \frac{\delta}{n}$ and ensuring that the widths of its bounds are smaller than $\frac{\epsilon}{2}$. For concreteness, we consider a randomized implementation of Direct, which we denote Uniform (and equivalently, as *policy* π_1).

> $\underline{\pi_1}$: Among arms with bound widths greater than $\frac{\epsilon}{2}$ when $\delta_{(i)} = \frac{\delta}{n}$, pull an arm uniformly randomly.

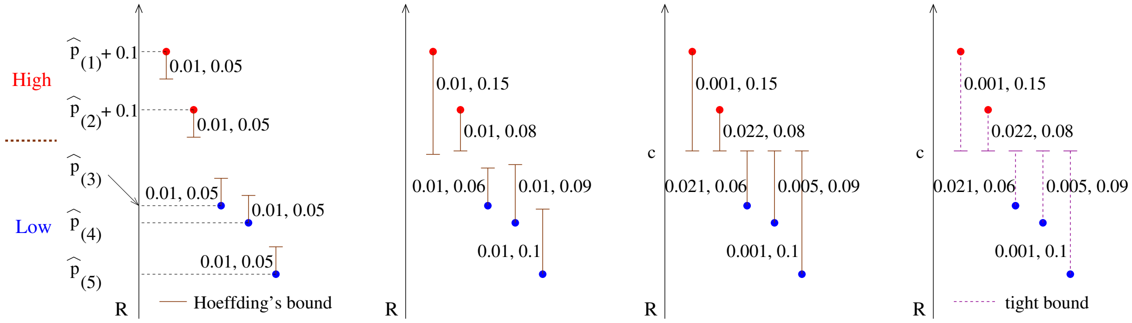
Figure 1 illustrates the evolution of our sampling policies on an instance of Explore-m with $n = 5$, $m = 2$, $\epsilon = 0.1$, $\delta = 0.05$. Figure 1(a) shows a schematic description of the various arms and their bounds when π_1 terminates. By being no larger than $\frac{\epsilon}{2}$, the widths of the bounds are *sufficient* for a PAC guarantee. The question that arises in immediate response is: what is *necessary* of the bounds in order to uphold a PAC guarantee? Indeed we observe that as long as arms from $High$ and Low do not collide, the PAC constraint would still be met. Correspondingly we update our policy to only select from among arms that are currently colliding; we denote the resulting policy π_2.

> $\underline{\pi_2}$: Among arms that collide when $\delta_{(i)} = \frac{\delta}{n}$, pull an arm uniformly randomly.

By only picking arms that collide, π_2 conserves samples in comparison to π_1. A key step in further improving sample efficiency follows from the observation that although bounds have different widths when π_2 terminates (Figure 1(b)), they are all computed under the same value $\delta_{(i)} = \frac{\delta}{n}$. Recall that we do not need the $\delta_{(i)}$ to be equal; only that their sum not exceed δ. In practice we find that significant economy can be achieved by allocating "larger portions of δ" to the arms that need it: arms with true means close to the boundary between $High$ and Low ((2) and (3) in Figure 1). We posit that arms with true means farther away from the boundary ((1) and (5)) would require relatively fewer samples to shrink their bounds enough to avoid collisions, even for low values of $\delta_{(i)}$.

Under both π_1 and π_2 we fix $\delta_{(i)}$ *a priori* such that their sum does not exceed δ, and sample arms enough times to eliminate collisions. By contrast, under our third policy, π_3, we manipulate the $\delta_{(i)}$ at every stage such that no collisions occur. Specifically we pick a "cutoff" c such that for every arm h in $High$, $LB_h = c$, and for every arm l in Low, $UB_l = c$ (Figure 1(c)). For each arm (i) we set $\delta_{(i)} = \delta^c_{(i)}$ such that its relevant upper or lower bound coincides with c. The quantity $\delta^c_{(i)}$ is obtained by *inverting* Hoeffding's inequality, and bounds the probability that the mean of arm (i) violates the cutoff c. $\delta^c_{(i)}$ effectively signifies how hard it is to make arm (i) collision free eventually: arms with high $\delta^c_{(i)}$ are likely to need more samples. We translate this intuition into policy π_3, which samples arm (i) with a probability proportional to $\delta^c_{(i)}$. Empirical results (Section 5) confirm that π_3 achieves a

[2] We say an arm in *High collides* if its lower bound is lower than the upper bound of some arm in *Low*; collision is defined similarly for arms in *Low*.



(a) π_1 (Uniform) (b) π_2 (c) π_3 (d) π_4 (Adapt)

Figure 1. Illustrative example of Explore-m ($n = 5$, $m = 2$, $\epsilon = 0.1$, and $\delta = 0.05$) showing progressively efficient sampling policies. The figure shows examples of terminal conditions when policies π_1 through π_4 are employed. Beside each bound is shown "$\delta_{(i)}$, $width_{(i)}$": if arm (i) is in $High$, $width_{(i)} = (\hat{p}_{(i)} + \epsilon) - (LB_{(i)} + \epsilon)$, and if it is in Low, $width_{(i)} = UB_{(i)} - \hat{p}_{(i)}$. The depicted values of $\delta_{(i)}$ and $width_{(i)}$ are not precise or drawn to scale; they are illustrative.

significant reduction in sample complexity over π_2.

$\underline{\pi_3}$: Pull arm (i) with a probability proportional to $\delta^c_{(i)}$.

Just as $\delta^c_{(i)}$ indicates arm (i)'s relative need for samples, the aggregate bound $\delta^c = 1 - \prod_{i=1}^{n}(1 - \delta^c_{(i)})$ summarizes the progress made by our sampling policy. For a given value of ϵ, after every pull we can compute δ^c as a bound on the probability that our current answer is incorrect; we can stop if $\delta^c \leq \delta$. Notice that indeed we can compute $\delta^c_{(i)}$ and δ^c at every stage for *any* sampling policy, including π_1 and π_2.

Naturally the choice of the cutoff c affects δ^c. Since it functions as a lower bound for arms in $High$ and as an upper bound for arms in Low, c must lie in the interval $I^{(m)}_{(m+1)} = (\hat{p}_{(m+1)}, \hat{p}_{(m)} + \epsilon)$, as in Figure 1(c). Indeed if each $\delta^c_{(i)}$ is derived using Hoeffding's inequality, we find that it is possible to efficiently compute $c^* = \operatorname{argmin}_{c \in I^{(m)}_{(m+1)}} \delta^c$. This computation relies on convexity properties of Hoeffding's bound. Searching for an optimal cutoff becomes less attractive when other bounds are used to compute $\delta^c_{(i)}$, as we observe shortly. For our experiments we use a cutoff $\hat{c}$ such that it divides the interval $I^{(m)}_{(m+1)}$ in proportion to the standard errors of the mean of arms $(m+1)$ and (m). We borrow this strategy from Chen et al. (2008), who consider a similar subset selection problem. Although easy to compute, $\hat{c}$ is a sub-optimal cutoff, as it only depends on arms (m) and $(m+1)$, whereas c^* includes terms from *all* the arms.

Our transition from π_3 to a policy π_4 is simple, yet practically significant. Rather than relying exclusively on Hoeffding's bound at every stage, we consider the tightest of the bounds that apply (Figure 1(d)). In particular, the recently proposed "empirical Bernstein bound" (Mnih et al., 2008) incorporates empirical variances into the calculation of bounds, and is particularly effective when some arms have relatively small variances.

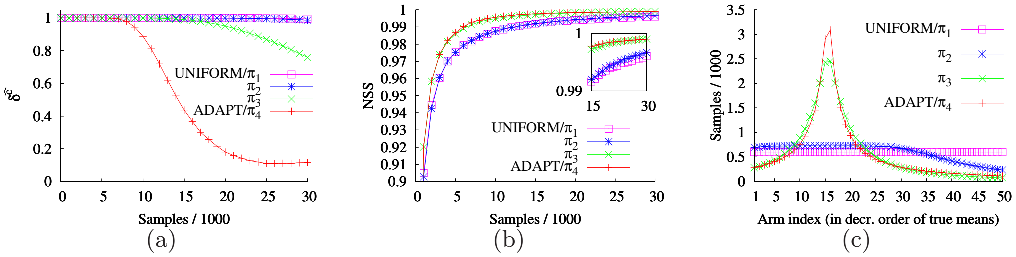
$\underline{\pi_4}$: Implement π_3 using the tightest applicable bounds to compute $LB_{(i)}$ and $UB_{(i)}$.

We refer to our final policy, π_4, as "Adapt". Although Adapt is extremely sample efficient in practice (as we see in the following section), and is indeed a PAC algorithm, it does not have a provably bounded *worst case* sample complexity. However, a worst case bound can be easily enforced by overlaying Adapt with a rule not to sample any arm more times than the Direct algorithm would. With more careful monitoring through rounds in our Halving algorithm, we could restrict Adapt to a worst case sample complexity of $O(\frac{n}{\epsilon^2} log(\frac{m}{\delta}))$. In our experiments we do not find such a need, as the PAC guarantee is typically realized early.

5. Experimental Results

In this section we compare various policies for Explore-m on a test instance with $m = 50$, $n = 15$, $\epsilon = 0.1$, and $\delta = 0.15$. The arms all have their true means drawn uniformly from the interval $[0, 1]$; each arm implements a uniform distribution with a standard deviation of 1 (a width of $\sqrt{12}$), which is relatively large compared to the spacing between the means. We report statistics on this instance of Explore-m based on averages of at least 1000 independent trials. In our comparisons the Adapt algorithm (π_4) predominantly achieves the best results with the fewest samples.

ADAPT: Figure 2 summarizes statistics comparing policies π_1 through π_4. Figure 2(a) plots the value of $\delta^{\hat{c}}$ against the number of pulls. The order among the policies is consistent with our expectation: notice that only π_4 is able to achieve the desired threshold of $\delta^{\hat{c}} \leq \delta = 0.15$ within the 30,000 samples plot-



(a) (b) (c)

Figure 2. Comparison of policies π_1 through π_4 on test instance ($n = 50$, $m = 15$, $\epsilon = 0.1$, $\delta = 0.15$). For each policy, (a) shows $\delta^{\hat{c}}$ and (b) shows NSS as a function of the number of pulls. In (c) arms $1, 2, \ldots, 50$ are in decreasing order of their *true* means, and a histogram shows the number of times each arm has been sampled. Notice peaks around 15 (m) and 16 ($m + 1$) under policies π_3 and π_4.

ted. While π_4 takes roughly 22,000 pulls to deliver the PAC guarantee, straightforward implementations of Direct (Algorithm 1) and Halving (Algorithm 3) would take millions of pulls at the same ϵ and δ values.

While $\delta^{\hat{c}}$ determines the PAC guarantee for choosing m (ϵ, m)-optimal arms, we evaluate the actual set of arms that would be returned at any stage based on the sums of their true means, i.e., $\sum_{i=1}^{m} p_{(i)}$. We normalize this "sum of selected means" such that it lies in $[0, 1]$. We denote the normalized sum NSS: note that the set T_m will have $NSS = 1$. Figure 2(b), which plots NSS against samples, shows a clear gulf between π_3 and π_4 on the one hand, and π_1 and π_2 on the other. Even after 30,000 pulls, this gap is significant ($p < 0.0001$).

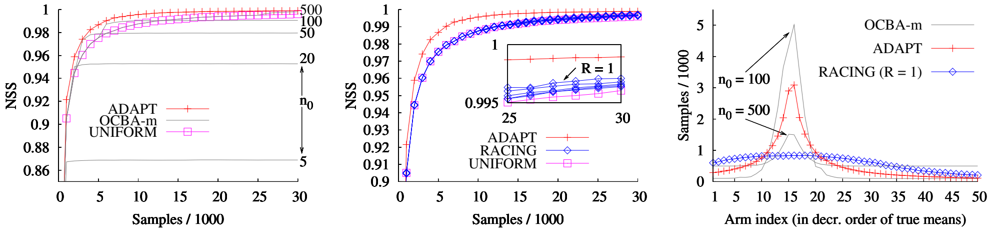
How do policies π_1 through π_4 ration samples among the arms? Figure 2(c) shows a histogram plotting the number of pulls as arms are sorted in decreasing order of their true means. As we proceed from π_1 to π_4 a growing bias towards "contentious" arms (on either side of m and $m+1$) is apparent. Recall that the arms have equal variances; this leads to a fairly symmetric pattern about m and $m + 1$ for policies π_3 and π_4.

OCBA-m: Chen et al. (2008) derive a sampling method similar to Adapt by adopting a Bayesian perspective. Under the "Optimal Computing Budget Allocation" (OCBA-m) framework, they model the true means of the arms with non-informative prior distributions, which get refined as samples are collected. First, each arm is sampled a fixed number (n_0) of times to obtain an estimate of its variance (see Chen et al., 2008, pp. 584–585). Subsequently arm (i) is allotted $\frac{s^2_{(i)}}{(\hat{p}_{(i)} - \hat{c})^2}$ samples, where $s_{(i)}$ is the sample standard deviation of arm (i), and $\hat{c}$ is the cutoff described in Section 4. Sampling is performed in batches of size Δ: only arms that have not already been sampled their allotted number are sampled. Consistent with our fully sequential approach in Adapt, we set $\Delta = 1$ in our implementation of OCBA-m, picking arms probabilistically in proportion to their allotments.

Clearly Adapt and OCBA-m are alike in spirit, focusing their attention on arms with higher variances and those close to the cutoff. However, while Adapt provides PAC bounds for subset selection, the Bayesian formulation under OCBA-m forces the use of approximations that are only valid in the asymptotic case. Indeed we find in our experiments that the performance of OCBA-m depends crucially on n_0, the number of rounds of uniform sampling. Chen et al. (2008) recommend setting $n_0 \geq 5$, and use $n_0 = 20$ in their experiments; however, for such low values we notice premature plateauing on our test instance (Figure 3(a)). For higher values of n_0 (100, 500), OCBA-m coincides with Uniform until a large number of pulls, and then outperforms it for a period when it switches to intelligent sampling.

We postulate that ideally the parameter n_0 must adapt to the relative spacing between arms and their variances, which it is not always possible to perceive *a priori*. In informal experiments, noting that OCBA-m does not have a tolerance parameter like ϵ, we compare the methods by setting $\epsilon = 0$ (PAC bounds are still possible under Adapt as long as the true means are separated). Additionally we set Gaussian distributions for the bandit arms with a standard deviation of 1 (for its bounds Adapt assumes that the support of this distribution is limited to six standard deviations). Although we do not report extensive results from these variations here, in these experiments Adapt still outperforms OCBA-m, for all the n_0 values reported in Figure 3(a), after $30,000$ samples ($p < 0.0001$).

RACING: In recent work Heidrich-Meisner & Igel (2009) extend the idea of racing algorithms (Maron & Moore, 1997) to the subset selection problem. Their algorithm ("Racing") proceeds for a *predetermined* number of rounds R. In each round arms are *selected* if with sufficient confidence they have higher means than $n - m$ others; they are *discarded* if they have lower means than m others. Such elimination is meant to progressively focus sampling on contentious arms.



(a) (b) (c)

Figure 3. (a) Comparisons with OCBA-m on our test instance. The plot shows NSS against the number of samples. Under OCBA-m, curves are shown for multiple values of n_0. (b) Similar curves for Racing, with multiple values of R (only $R = 1$ is marked). (c) A histogram showing the number samples allocated to the bandit arms by the algorithms.

Like Adapt, Racing can also provide a PAC guarantee of choosing the best subset (there is no tolerance parameter ϵ). However, the guarantee can only be provided if m arms have indeed been selected at the end of R rounds. Herein arises a dilemma: setting R to a high value could provide more rounds for sampling; however, the permissible mistake probabilities while eliminating arms are (roughly) inversely proportional to R. Racing has to enforce conservative mistake probabilities because elimination is an irrevocable decision, and further, bounds on the true means of arms are not allowed to expand from round to round.

On our test instance, we are unable to select a subset of size m within the specified number of rounds R despite trying multiple values (between 1 and 500). In response, we continue to run the algorithm (see Heidrich-Meisner & Igel, 2009, p. 404) beyond R rounds, using the confidence probabilities prescribed for round R. Not surprisingly, the best NSS results are achieved when we set $R = 1$ (Figure 3(b)), which provides the largest scope for eliminating arms. As R is increased (shown, but not marked in the figure), Racing progressively tends towards the Uniform algorithm, since fewer and fewer arms get eliminated. We posit that Racing is likely to perform better on a smaller problem instances, a hypothesis that is confirmed when we set $n = 10$, $m = 3$, $\epsilon = 0$, and $\delta = 0.15$. Indeed Racing ($R = 1$) achieves a higher NSS value after $30,000$ pulls compared to Uniform, and is *not* distinguishable from Adapt ($p < 0.05$).

By and large, we observe that the superior sample efficiency of Adapt is most pronounced when the number of arms n is large, and the arms have relatively high variances. Clear differences in the sampling strategies of Adapt, OCBA-m, and Racing are visible in Figure 3(c), which plots histograms of their pulls. OCBA-m and Adapt are both more peaked than Racing. However, we notice that Adapt has a gradual slope on either side of its peak, while OCBA-m rises more dramatically near the separating boundary.

6. Related Work and Discussion

In the previous sections we have already encountered work that is very closely related to our contribution; we conclude by highlighting some further connections. Our exploration strategy in the multi-armed bandit attempts to minimize the number of samples to achieve a prescribed level of confidence about the selected subset. By contrast, alternative formulations for exploration in bandits consider strategies to maximize the chances of identifying the best arm when provided a *fixed* budget of samples (Madani & Lizotte, 2004). Even-Dar et al. (2006) extend their analysis of Explore-1 to develop exploration strategies for MDPs, resulting in reinforcement learning algorithms with similar PAC guarantees. In general an optimal policy for an MDP only relies on a single optimal action for every state, and so it appears unlikely that our generalization to Explore-m will have a direct bearing on their problem. However, any practical implementation of their algorithm could benefit from using Adapt, which applies equally to the case when $m = 1$.

Subset selection has been widely studied under varying assumptions. In early work Koenig & Law (1985) provide a two-stage sampling procedure for selecting a subset of size m containing the l best of k independent normal populations. Kim & Nelson (2001) formalize the notion of an "indifference zone" in subset selection, which corresponds closely to the parameter ϵ in Explore-m. In their survey of empirical methods for choosing the single best candidate from a population, Inoue et al. (1999) suggest that Bayesian approaches, which address the average case, could be more beneficial in practice than worst case formulations. Through Adapt we contribute an algorithm that effectively manipulates bounds to achieve efficiency in practice while preserving a PAC guarantee.

Efficient subset selection is a primary subroutine in several evolutionary algorithms (Schmidt et al., 2006) and stochastic optimization methods (de Boer et al., 2005). We expect that Adapt, which is straightforward to implement, can easily be integrated into existing code bases to improve sample efficiency. However, it must be noted that while Explore-m specifically implements subset selection, in general evolutionary algorithms could employ other selection schemes, such as tournament and proportionate selection (Miller & Goldberg, 1996). Also, some evolutionary algorithms seek to maintain good *on-line* performance (Whiteson & Stone, 2006), which the "pure exploration" nature of Explore-m does not match.

The tolerance parameter ϵ in Explore-m can naturally control the tradeoff between the quality of selections on each iteration of an evolutionary algorithm and the algorithm's overall sample efficiency. Indeed ϵ is the only input parameter of Adapt, which seeks to reduce its mistake probability $\delta^{\hat{c}}$ with each sample. Note that OCBA-m also provides a probabilistic measure similar to $\delta^{\hat{c}}$ to gauge the algorithm's progress; in contrast, Racing only preserves its PAC guarantee for a fixed number of rounds. As our experiments show, the effects of the input parameters to Racing (R) and OCBA-m (n_0) are not easy to intuit. We believe that the superior empirical performance of Adapt, supported by its ease of use and principled theoretical grounding, make it a promising method for the fundamental problem of subset selection.

Acknowledgments

We thank Todd Hester, Tobias Jung, Juhyun Lee, Shimon Whiteson, and the anonymous reviewers of this paper for their useful suggestions. This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-0615104 and IIS-0917122), ONR (N00014-09-1-0658), DARPA (FA8650-08-C-7812), and the Federal Highway Administration (DTFH61-07-H-00030).

References

Berry, Donald A. and Fristedt, Bert. *Bandit problems*. Chapman and Hall Ltd., 1985.

Chen, Chun-Hung, He, Donghai, Fu, Michael, and Lee, Loo Hay. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS Journal on Computing*, 20(4):579–595, 2008.

de Boer, Pieter-Tjerk, Kroese, Dirk P., Mannor, Shie, and Rubinstein, Reuven Y. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

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

Heidrich-Meisner, Verena and Igel, Christian. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *Proc. ICML 2009*, pp. 401–408. ACM, 2009.

Inoue, Koichiro, Chick, Stephen E., and Chen, Chun-Hung. An empirical evaluation of several methods to select the best system. *ACM Transactions on Modeling and Computer Simulation*, 9(4):381–407, 1999.

Kim, Seong-Hee and Nelson, Barry L. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11(3):251–273, 2001.

Koenig, Lloyd W. and Law, Averill M. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. *Communications in statistics. Simulation and computation*, 14(3):719–734, 1985.

Madani, Omid and Lizotte, Daniel J. Greiner, Russell. Active model selection. In *Proc. UAI 2004*, pp. 357–365. AUAI Press, 2004.

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

Maron, Oded and Moore, Andrew W. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1–5):193–225, 1997.

Miller, Brad L. and Goldberg, David E. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2):113–131, 1996.

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

Schmidt, Christian, Branke, Jürgen, and Chick, Stephen E. Integrating techniques from statistical ranking into evolutionary algorithms. In *Appl. of Evolutionary Comp.*, volume 3907 of *LNCS*, pp. 752–763. Springer, 2006.

Whiteson, Shimon and Stone, Peter. On-line evolutionary computation for reinforcement learning in stochastic domains. In *Proc. GECCO 2006*, pp. 1577–1584. ACM, 2006.