PAC Algorithms for the Infinitely-Many Armed Problem with Multiple Pools

Yahel David yaheld@tx.technion.ac.il
Nahum Shimkin shimkin@ee.technion.ac.il
Faculty of Electrical Engineering
Technion, Haifa 32000, Israel

Editor:

Abstract

We consider a multi-pool version of the infinitely-many armed bandit problem, where a learning agent is faced with several large pools of items, and interested in finding the best item overall. At each time step the agent chooses a pool, and obtains a random item whose value is precisely revealed. The obtained values within each pool are assumed to be i.i.d., with an unknown probability distribution that generally differs among the pools. Under the PAC framework, we provide lower bounds on the sample complexity of any (ϵ, δ)-correct algorithm, and propose an algorithm that attains this bound up to logarithmic factors. We compare the performance of this multi-pool algorithm to the variant in which the pools are not distinguishable by the agent and are chosen randomly at each stage. Interestingly, when the supremal values of the pools happen to be similar, the latter approach may provide better performance.

Keywords: Multi-armed bandits, pure exploration.

1. Introduction

We consider the problem of finding the best item from a large set of items, or arms, which are arranged in separate pools. The value distributions of the items in each pool is unknown, but may well be different across pools. A learning agent samples the pools sequentially, where at each time step it chooses some pool, obtains a random element from that pool, and observes its numerical value.

The goal of the agent is to quickly return an (ϵ, δ)-correct arm, namely an arm whose value is ϵ-close to the overall best arm with a probability larger than $1 - \delta$. Specifically, we wish to minimize the sample complexity, namely the expected number of samples observed by the learning algorithm before it terminates. Our model assumes that the value of each newly sampled arm is an independent sample from a pool-dependent probability distribution. We further assume that the probability distribution function of each pool is continuous, and has a density which is bounded from below, with a known lower bound.

The scenario considered here is relevant when a single item needs to be selected from among several clustered sets. These may be parts that come from different manufacturers or produced by different processes, employees that are refereed by different employment agencies, finding the best match to certain genetic characteristics in different populations, or choosing the best channel among different frequency bands in a cognitive radio wireless

system. We note that our model considers the pools as being initially indistinguishable, in the sense that no prior knowledge is presumed regarding their relative merit.

The model considered here is related to the so-call infinitely-many armed bandit problem, studied by Berry et al. (1997); Bonald and Proutiere (2013); Chakrabarti et al. (2009); David and Shimkin (2014); Teytaud et al. (2007); Wang et al. (2008). These works consider the case of single pool, focusing on online learning with the regret criterion. In most of these works the observed values are stochastic, so that repeated sampling of each observed arm is generally required to learn its value. Here, we assume that the values of the arms are fixed and precisely revealed once sampled, which enables us to focus on the choice of pool as the main decision issue.

For the classical Multi-Armed Bandit (MAB) problem, algorithms that find the best arm (in terms of its expected value) in the PAC sense sense were presented by Even-Dar et al. (2002). For the same problem, a lower bound on the sample complexity was presented by Mannor and Tsitsiklis (2004). Our model can be viewed as analogous to this MAB model by considering, respectively, the pools as the arms, and the item values in our model as the stochastic rewards in the MAB problem. The essential difference is in the objective, which in our case is to find and retain the item (i.e., sample) with the highest value.

From another perspective, the proposed model may be compared to the *secretary problem*, see for example Babaioff et al. (2008); Freeman (1983) for extensive surveys. In that problem the goal is to maximize the probability of hiring the best candidate in a finite group, while in our model we seek an ϵ-optimal item from an unlimited set of items, which is further divided among different pools.

The paper proceeds as follows. In the next section we present our model. In Section 3 we provide a lower bound on the sample complexity of every (ϵ, δ)-correct algorithm. In Section 4 we present an (ϵ, δ)-correct algorithm, and we provide an upper bound on its sample complexity which has the same order of the lower bound up to a logarithmic term in the number of pools. In Section 5, we consider for comparison the single-pool variant of the model, where the agent cannot distinguish between pools and samples from them uniformly at random.
CONCLUDING REMARKS

2. Model

We consider a set of pools, denoted by K. When the learning agent is sampling from pool $k \in K$, a value or *arm* is revealed. We assume that the arms' values which are obtained from pool $k \in K$ are distributed according the probability density function (*p.d.f.*) $f_k(\mu)$ and we denote the cumulative distribution function (*c.d.f.*) of this pool by $F_k(\mu)$. We assume that $f_k(\mu) \geq a$, $\forall k \in K$, for some known constant $a > 0$ and that $F_k(\mu)$ is continuous. We denote the support of $f_k(\mu)$ by $supp(f_k)$. We assume that $supp(f_k)$ is a single interval, that $supp(f_k) \in [0, 1]$ and we denote its supremal value by $\mu_k^* = \sup\{\mu | \mu \in supp(f_k)\}$. The largest value among all of the pools is denoted by $\mu_*^* = \max_{k \in K} \mu_k^*$.

An algorithm for this model, selects a pool, samples from it and receives an arm at each time step. For making its selection, the algorithm may relay on the history (i.e. the actions and the obtained arms). An algorithm is (ϵ, δ)-correct if

$$P\left(I > \mu_*^* - \epsilon\right) > 1 - \delta$$

where I stands for the value of the arm that the algorithm provides at the end of the running. The expected number of samples after which the algorithm terminates is called the *sample complexity* of the algorithm, the number of samples is denoted by T.

3. Lower Bound

In this section we present a lower bound on the sample complexity.

Theorem 1 *For every $\epsilon \in (0, \frac{1}{2})$ and $\delta \in (0, \delta_0)$ where $\delta_0 < \frac{3}{16}$, for the case in which $\mu_*^* \leq \frac{1}{2}$, for every (ϵ, δ)-correct policy, we have*

$$E[T] \geq \min_{k' \in K} \sum_{k \in K \setminus \{k'\}} \frac{1}{8a\left(\epsilon + \mu_*^* - \mu_i^*\right)} \ln\left(\frac{3}{16\delta}\right). \tag{1}$$

Note that the above Theorem can be generalized to the case in which $\epsilon \in (0, \epsilon_0)$ and $\mu_*^* \leq 1 - \epsilon_0$, where $\epsilon_0 < 1$.

For proving the above Theorem we show that if an algorithm is (ϵ, δ)-correct and that it's sample complexity is lower than a certain threshold for some probabilities functions of the pools, then this algorithm will not be (ϵ, δ)-correct for another probabilities functions of the pools.

Proof First , we define the following hypotheses:

$$H_0: \quad f_k^{H_0}(\mu) = f_k(\mu) \quad \forall k \in K,$$

and for every $k \in \{1, ..., |K|\}$

$$H_k: \quad f_k^{H_k}(\mu) = \max\left(\gamma_k f_k(\mu), a\right) \mathbf{1}_{supp(f_k)}(\mu) + a\mathbf{1}_{\left[\mu_k^*, \mu_*^* + \epsilon\right]}(\mu), \quad f_l^{H_k}(\mu) = f_l(\mu),\ \forall l \in \{l \in K | l \neq k\},$$

where $supp(f_k)$ stand for the support of $f_k(\mu)$ and γ_k is chosen such that $\int_0^1 f_k^{H_k}(\mu)d\mu = 1$. For bounding γ_k we note that for

$$g_k(\mu) \triangleq \gamma_k' \left(f_k(\mu) - a\mathbf{1}_{supp(f_k)}(\mu)\right) + a\mathbf{1}_{supp(f_k)}(\mu) + a\mathbf{1}_{\left[\mu_k^*, \mu_*^* + \epsilon\right]}(\mu),$$

where γ_k' is chosen such that $\int_0^1 g_k(\mu)d\mu = 1$ it follows that $\gamma_k' \leq \gamma_k$. Then by the fact that

$$\int_0^1 g_k(\mu)d\mu = \gamma_k'\left(1 - a|supp(f_k)|\right) + a|supp(f_k)| + a\left(\mu_*^* + \epsilon - \mu_k^*\right),$$

where $|supp(f_k)| = \int_0^1 \mathbf{1}_{supp(f_k)}(\mu)d\mu$ and by the fact that $\int_0^1 g_l(\mu)d\mu = 1$, it is obtained that

$$1 - \frac{a\left(\mu_*^* + \epsilon - \mu_k^*\right)}{1 - a|supp(f_k)|} \leq \gamma_k' \leq \gamma_k \leq 1.$$

If hypothesis H_k is true, the algorithm should provide a value from pool k. We use E_k and P_k to denote the expectation and probability respectively, under the policy being considered and under hypothesis H_k. Now, for every $k \in K$ let

$$t_k = \frac{1}{8a\left(\epsilon + \mu_*^* - \mu_k^*\right)} \ln\left(\frac{3}{16\delta}\right),$$

and let T_k stands for the number of samples from pool k.

Now, we assume we run a policy which is (ϵ, δ)-correct under H_0. We will show that if under this policy $E[T_k] \leq t_k$, then, this policy can't be (ϵ, δ)-correct under hypothesis H_k. Therefore, an (ϵ, δ)-correct policy must have $E[T_k] > t_k,\ \forall k \in K$.

First, we define the event $A_k = \{T_k \leq 4t_k\}$. It easily follows by

$$4t_k \left(1 - P_0(A_k)\right) \leq E_0[T_k],$$

that if $E_0[T_k] \leq t_k$, then $P_0(A_k) \geq \frac{3}{4}$.

Let B_k stand for the event under which the chosen sample is from pool k, and B_k^C for its complementary. Since at most for one pool $k' \in K$ it can be obtained that $P_0\left(B_{k'}\right) > \frac{1}{2}$, it follows that $P_0\left(B_k^C\right) > \frac{1}{2}$ for every $k \in K \setminus \{k\}$.

We define the event C_k to be the event under which all the samples obtained from pool k are on the interval $[0, \mu_k^*]$. Clearly, $P_0(C_k) = 1$.

We define the event $S_k = \{A_k \cap B_k^C \cap C_k\}$, since we have already shown that for every $k \in K \setminus \{k'\}$, $P_0(A_k) \geq \frac{3}{4}$, $P_0(B_k^C) > \frac{1}{2}$ and $P_0(C_k) = 1$ it is obtained that $P_0\left(S_k\right) > \frac{3}{8},\ \forall k \in K \setminus \{k'\}$. Then, since for every history of N samples, for which the event C_k holds, it is obtained that $\frac{dP_k}{dP_0} \geq \gamma_k^N$, we have the following,

$$P_k\left(B_k^C\right) \geq P_k\left(S_k\right) = E_0\left[\frac{dP_k}{dP_0} I\left(S_k\right)\right] \geq \gamma_k^{-4t_k} P_0\left(I\left(S_k\right)\right) > \frac{3}{8}\gamma_l^{-4t_k} \geq \frac{3}{16} e^{-\frac{1}{2(1-a)} \ln \frac{3}{16\delta}} \geq \delta,$$

where in the last inequality we used the facts that $(1-\epsilon)^{\frac{1}{\epsilon}} \geq e^{-1}$ and that $|supp(f_k)| \leq 1$.

We found that if a policy is (ϵ, δ)-correct under hypothesis H_0 and $E_0[T_k] \leq t_k$ for some $k \neq k'$, then, under hypothesis H_k this policy is not (ϵ, δ)-correct. So, for having an (ϵ, δ)-correct policy, we must have $E_0[T_k] > t_k$ for all of the arms except the one for which $P_0\left(B_k^C\right) \leq \frac{1}{2}$. Hence the lower bound is obtained. ■

4. Algorithm

Here we provide an (ϵ, δ)-correct algorithm. This algorithm samples from each pool once. Then, it repeatedly calculates an upper bound on the supremal value of each pool and samples one arm from the pool for which the bound is the largest. The algorithm terminates when the number of samples from the chosen pool is above a certain threshold. This idea is similar to that in the UCB1 Algorithm provided by Auer et al. (2002).

Theorem 2 *For $L \geq 10$, Algorithm 1 is (ϵ, δ)-correct with sample complexity of*

$$E[T] \leq \sum_{k \in K} \frac{L - \ln(\delta)}{a \max\left(\epsilon, \mu_*^* - \mu_i^*\right)} + |K| + 1,$$

where $L = 6 \ln\left(|K|\left(1 + \frac{-\ln(\delta)}{a\epsilon}\right)\right)$ is defined in the algorithm.

Note that L is logarithmic in $|K|$. Hence, the upper bound on the sample complexity is of the same order as the lower bound in Theorem 1, up to a logarithmic factor in $|K|$.

Algorithm 1 Multi Pool Algorithm

1: **Input:** Constants $\delta > 0$, $\epsilon > 0$ and $L = 6\ln\left(|K|\left(1 + \frac{-\ln(\delta)}{a\epsilon}\right)\right)$.
2: **Initialization:** Counters $C(i) = 1\,\forall i \in K$.
3: Sample one arm from every pool.
4: Compute $Y^i_{C(i)} = V^i_{C(i)}(1) + \epsilon^{UB}(C(i))$ where $\epsilon^{UB}(C(i)) = \frac{L-\ln(\delta)}{aC(i)}$ and set $i^* = \arg\max_{i\in K} Y^i_{C(i)}$.
5: If $\epsilon^{UB}(C(i^*)) < \epsilon$, return the best sampled arm.
Else, sample one arm from pool i^*, set $C(i^*) = C(i^*) + 1$ and return to step 4.

For proving Theorem 2, we first bound the probability of the event under which the upper bound of the best pool is below the supremal value. Then, we bound the largest number of samples after which the algorithm terminates under the assumption that the upper bound of the best pool is above the supremal value.

Proof First we denote the time step of the algorithm by t, and the value of the counter $C(i)$ at time step t by $C^t(i)$. Recall that T stands for the random final time step. By the condition in step 5 of the algorithm, for every pool $k \in K$, it follows that,

$$C^T(k) \leq \lfloor \frac{L - \ln(\delta)}{a\epsilon} \rfloor + 1. \tag{2}$$

Note that by the fact that for $x \geq 6$ it follows that $\frac{d6\ln(x)}{dx} \leq 1$, and by the fact that for $x_0 = \exp\left(1\frac{2}{3}\right)$ it follows that $x_0 > 6\ln(x_0) = 10$ it is obtained that

$$L' \triangleq |K|\left(\frac{-\ln(\delta)}{a\epsilon} + 1\right) > \ln\left(|K|\left(\frac{-\ln(\delta)}{a\epsilon} + 1\right)\right) = L,$$

for $L \geq 10$. So, by the fact that $T = \sum_{i\in K} C^T(i)$, for $L \geq 10$ it follows that

$$T \leq |K|\left(\frac{L - \ln(\delta)}{a\epsilon} + 1\right) < |K|\left(\frac{L' - \ln(\delta)}{a\epsilon} + 1\right) \leq L'^2 = e^{\frac{L}{3}}. \tag{3}$$

Now, we begin with proving the (ϵ, δ)-correctness property of the algorithm. Recall that for every pool $k \in K$ the values are distributed according to the c.d.f. $F_k(\mu)$. Let assume w.l.o.g. that $\mu^*_1 = \mu^*_*$. Then, for $N > 0$ and by the fact that $(1-\epsilon)^{\frac{1}{\epsilon}} \leq e^{-1}$ for every $\epsilon \in (0, 1]$, for $\epsilon^{UB}(N) = \frac{L-\ln(\delta)}{Na}$ it follows that

$$P\left(V^1_N < \mu^*_* - \epsilon^{UB}(N)\right) = \left(F\left(\mu^*_* - \epsilon^{UB}(N)\right)\right)^N \leq \left(1 - a\epsilon^{UB}(N)\right)^N \leq \delta e^{-L}. \tag{4}$$

Hence, at every time step t, by the definition of $Y^1_{C^t(1)}$ and Equations (3) and (4), by applying the union bound, it follows that

$$P\left(Y^1_{C^t(1)} < \mu^*_*\right) \leq P\left(V^1_{C^t(1)} < \mu^*_* - \epsilon^{UB}(C^t(1))\right) \leq \sum_{t=1}^{\exp\left(\frac{L}{3}\right)} P\left(V^1_N < \mu^*_* - \epsilon^{UB}(N)\right) \leq \delta e^{-\frac{2L}{3}}. \tag{5}$$

Since by the condition in step 5, it is obtained that when the algorithm stops

$$V_{C^t(i^*)}^{i^*} > Y_{C^t(i^*)}^{i^*} - \epsilon,$$

and by the fact that for every time step

$$Y_{C^t(i^*)}^{i^*} \geq Y_{C^t(1)}^{1},$$

it follows by Equation (5) that

$$P\left(V_{C^t(i^*)}^{i^*} \leq \mu_*^* - \epsilon\right) \leq P\left(Y_{C^t(1)}^{1} < \mu_*^*\right) \leq \delta e^{-\frac{2L}{3}}.$$

Therefore, it follows that the algorithm returns an arm greater than $\mu_*^* - \epsilon$ with a probability larger than $1-\delta$. So, it is (ϵ, δ)-correct.

For proving the bound on the expected sample complexity of the algorithm we define the following sets:

$$M(\epsilon) = \{l \in K | \mu_*^* - \mu_l^* < \epsilon\}, \quad N(\epsilon) = \{l \in K | \mu_*^* - \mu_l^* \geq \epsilon\}.$$

As before, we assume w.l.o.g. that $\mu_1^* = \mu_*^*$. For the case in which

$$E_1 \triangleq \bigcap_{1 \leq t < T} \left\{Y_{C^t(1)}^{1} \geq \mu_*^*\right\},$$

occurs, since $V_{C^t(k)}^{k} \leq \mu_k^*$ for every $k \in K$, and every time step, it follows that the necessary condition for sampling from pool k,

$$Y_{C^k(1)}^{k} \geq Y_{C^t(1)}^{1},$$

occurs only when the event

$$E_2(t) \triangleq \left\{\mu_k^* + \epsilon^{UB}\left(C^t(k)\right) \geq \mu_*^*\right\},$$

occurs. But

$$E_2(t) \subseteq \left\{C^t(i) \leq \frac{L - \ln(\delta)}{a\left(\mu_*^* - \mu_k^*\right)}\right\}.$$

Therefore, it is obtained that

$$C^T(i) \leq \lfloor \frac{L - \ln(\delta)}{a\left(\mu_*^* - \mu_k^*\right)} \rfloor + 1. \tag{6}$$

By using the bound in Equation (2) for the pools in the set $M(\epsilon)$, the bound in Equation (6) for the pools in the set $N(\epsilon)$ and the bound in Equation (3), it is obtained that

$$E[T] \leq \left(1 - P\left(E_1\right)\right) e^{\frac{L}{3}} + P\left(E_1\right) \Phi\left(\epsilon\right), \tag{7}$$

where

$$\Phi(\epsilon) \triangleq \left(\sum_{k \in N(\epsilon)} \left(\lfloor \frac{L - \ln(\delta)}{a\left(\mu_*^* - \mu_k^*\right)} \rfloor + 1 \right) + \sum_{k \in M(\epsilon)} \left(\lfloor \frac{L - \ln(\delta)}{a\epsilon} \rfloor + 1 \right) \right).$$

In addition, by Equation (5), the bound in Equation (3) and by applying the union bound, it follows that

$$P(E_1) \geq 1 - \sum_{t=1}^{T} P\left(Y_{C^t(1)}^1 < \mu_*^*\right) \geq 1 - \delta e^{-\frac{2L}{3}} e^{\frac{L}{3}} = 1 - \delta e^{-\frac{L}{3}}.$$

So,

$$1 - P(E_1) \leq \delta e^{-\frac{L}{3}}. \tag{8}$$

Furthermore, by the definitions of the sets $N(\epsilon)$ and $M(\epsilon)$, it can be obtained that

$$\Phi(\epsilon) \leq \sum_{k \in K} \lfloor \frac{L - \ln(\delta)}{a \max\left(\epsilon, \mu_*^* - \mu_k^*\right)} \rfloor + 1. \tag{9}$$

Therefore, by Equation (7), (8) and (9) the bound on the sample complexity is obtained. ■

5. Comparison with The Single Pool Model

In this section, we analyze the improvement in the sample complexity obtained by using the multi pool property (the fact we can choose from which pool to sample at each time step) compared to a model in which all the pools are considered as a single pool. In the single pool model, when the agent samples from this single pool, a certain pool (among the multi pool) is chosen uniformly and an arm is sampled from this pool. We denote the p.d.f. and the c.d.f. of the single pool as $f(\mu)$ and $F(\mu)$ respectively. By the definition of this pool and our assumption in Section 2, its obtained that $f(\mu) \geq \frac{a}{|K|}$ and that it's supremal value is μ_*^*.

For the problem of regret minimizing (or maximizing the cumulative reward) the single pool model, with fixed arms' values was studied by David and Shimkin (2014).

In the remainder of this section, we provide a lower bound on the sample complexity and an (ϵ, δ)-correct algorithm that attains the same order of this bound for the single pool model. Then, we discuss which approach (multi pool vs. single pool) is better for which case and provide examples that illustrate these cases.

5.1 Lower Bound

In the following Theorem we provide a lower bound on the sample complexity for the single pool model.

Theorem 3 *For every $\epsilon \in (0, \frac{1}{2})$ and $\delta \in (0, \delta_0)$ where $\delta_0 < \frac{3}{5}$, for the case in which $\mu_*^* \leq \frac{1}{2}$, for every (ϵ, δ)-correct policy, we have*

$$E[T] \geq \frac{|K|}{4a\epsilon} \ln\left(\frac{3}{5\delta}\right). \tag{10}$$

Algorithm 2 Single Pool Algorithm

1: **Input:** Constants $\delta > 0$, $\epsilon > 0$.
2: Sample $\lceil \frac{-\ln(\delta)|K|}{a\epsilon} \rceil + 1$ arms from the pool.
3: Return the best sampled arm.

The proof is provided in Appendix A and is based on the same idea as the proof of Theorem 1.

5.2 Algorithm

In Algorithm 2 a certain number of arms is sampled and then the algorithm choose the best one among them. In the following Theorem we provide a bound on the sample complexity achieved by Algorithm 2.

Theorem 4 *Algorithm 2 is (ϵ, δ)-correct with sample complexity of*

$$E[T] \leq \frac{-|K| \ln(\delta)}{a\epsilon} + 2.$$

Note that the upper bound on the sample complexity is of the same order as the lower bound in Theorem 3.

The proof is provided in Appendix B.

5.3 Comparison Between The Models and Examples

By the bound provided in Theorem 2 for Algorithm 1 and the lower bound for the single pool model provided in Theorem 3, it follows that using the multi pool property helps to eliminate sampling from pools which have much smaller supremal value compared to the best pool (in comparing to ϵ). Hence, in these cases it always better to use the multi pool property and applying Algorithm 1.

But, in cases in which all the pools have approximately the same supremal value (compared to ϵ), the performance of Algorithm 2 are better than those of Algorithm 1 since we have an additional logarithmic in $|K|$ multiplicity factor in the bound of Algorithm 2.

We now provide two examples which illustrate the above discussion. In the following example we take small ϵ, hence using the multi pool property will be very effective.

Example 1 *Let $|K| = 1000$, $\mu_1^* = 0.4$, $\mu_k^* = 0.1$, $\forall k \in K \setminus \{1\}$ and $a = 0.01$. Then, for $\epsilon = 0.001$ and $\delta = 0.001$ the sample complexity attained by Algorithm 1 is 5.58×10^7. This sample complexity is smaller than the lower bound on the sample complexity for the single pool model provided in Theorem 3, which is 1.59×10^8. The sample complexity attained by the Algorithm 2 (which does not use the multi pool property) is 6.9×10^8.*

In the next example we will take a larger ϵ, hence the logarithmic in $|K|$ multiplicity factor, which is the drawback of Algorithm 1, will be more effective than the advantage of using the multi pool property.

Example 2 *Let $|K|$, μ_1^*, μ_k^* $\forall k \in K \setminus \{1\}$, $a = 0.01$ and δ remain the same as in Example 1, and let $\epsilon = 0.1$. The sample complexity of Algorithm 1 is 3.8×10^7, which is larger than the sample complexity of Algorithm 2 which is 6.9×10^6.*

As explained before and illustrated in the above examples, Algorithm 1 has an additional multiplicity factor which is logarithmic in $|K|$. Hence for some values of ϵ the using of Algorithm 2 attains smaller sample complexity compared to Algorithm 1. But for any set of pools, for small enough ϵ Algorithm 1 always attains smaller sample complexity compared to Algorithm 2 and to the lower bound for the single pool model which is provided in Theorem 3.

Remark 5 *In some cases, the lowest sample complexity is achieved by merging every group of a certain number of pools into one pool, and then applying Algorithm 1. In the case of Example 1, by merging every two pools into one the bound on the sample complexity attained by Algorithm 1 is* 4.57×10^7.

References

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

Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Online auctions and generalized secretary problems. *ACM SIGecom Exchanges*, 7(2):1–11, 2008.

Donald A Berry, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, pages 2103–2116, 1997.

Thomas Bonald and Alexandre Proutiere. Two-target algorithms for infinite-armed bandits with Bernoulli rewards. In *Advances in Neural Information Processing Systems 26*, pages 2184–2192. Curran Associates, Inc., 2013.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Advances in Neural Information Processing Systems 21*, pages 273–280. Curran Associates, Inc., 2009.

Yahel David and Nahum Shimkin. Infinitely many-armed bandits with unknown value distribution. In *Machine Learning and Knowledge Discovery in Databases*, pages 307–322. Springer, 2014.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In Jyrki Kivinen and RobertH. Sloan, editors, *Computational Learning Theory*, volume 2375 of *Lecture Notes in Computer Science*, pages 255–270. Springer Berlin Heidelberg, 2002.

PR Freeman. The secretary problem and its extensions: A review. *International Statistical Review*, pages 189–206, 1983.

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

Olivier Teytaud, Sylvain Gelly, and Michèle Sebag. Anytime many-armed bandits. In *CAP*, Grenoble, France, 2007.

Yizao Wang, Jean-Yves Audibert, Rémi Munos, et al. Infinitely many-armed bandits. *Advances in Neural Information Processing Systems*, 8:1–8, 2008.

6. Appendix A

Proof [Theorem 3] First , we define the following hypotheses:

$$H_0: \quad f^{H_0}(\mu) = f(\mu),$$

and

$$H_1: \quad f^{H_1}(\mu) = \max\left(\gamma f(\mu), a\right) \mathbf{1}_{supp(f)}(\mu) + \frac{a}{|K|}\mathbf{1}_{[\mu_*^*, \mu_*^*+\epsilon]}(\mu),$$

where, as in the proof of Theorem 1, $supp(f)$ stand for the support of $f(\mu)$ and γ is chosen such that $\int_0^1 f^{H_1}(\mu)d\mu = 1$. Similarly to the proof of Theorem 1, for bounding γ we note that for

$$g(\mu) \triangleq \gamma'\left(f(\mu) - a\mathbf{1}_{supp(f)}(\mu)\right) + a\mathbf{1}_{supp(f)}(\mu) + \frac{a}{|K|}\mathbf{1}_{[\mu_*^*, \mu_*^*+\epsilon]}(\mu),$$

where γ' is chosen such that $\int_0^1 g(\mu)d\mu = 1$ it follows that $\gamma' \leq \gamma$. Then by the fact that

$$\int_0^1 g(\mu)d\mu = \gamma'\left(1 - a|supp(f)|\right) + a|supp(f)| + \frac{a\epsilon}{|K|},$$

where $|supp(f)| = \int_0^1 \mathbf{1}_{supp(f)}(\mu)d\mu$ and by

$$\int_0^1 g(\mu)d\mu = 1,$$

it is obtained that $1 - \frac{a\epsilon}{|K|(1-a|supp(f)|)} \leq \gamma' \leq \gamma \leq 1$.

If hypothesis H_1 is true, the algorithm should provide a value greater than μ_*^*. We use E_l and P_l (where $l \in \{0, 1\}$) to denote the expectation and probability respectively, under the policy being considered and under hypothesis H_l. Now, let

$$t = \frac{|K|}{4a\epsilon}\ln\left(\frac{3}{5\delta}\right),$$

and recall that T stands for the total number of samples from the pool.

Now, we assume we run a policy which is (ϵ, δ)-correct under H_0. We will show that if under this policy $E[T] \leq t$, then, this policy can't be (ϵ, δ)-correct under hypothesis H_1. Therefore, an (ϵ, δ)-correct policy must have $E[T] > t$.

First, we define the event $A = \{T \leq 4t\}$. By the same consideration as in the proof of Theorem 1 (for the events $\{A_k\}_{k\in K}$), it follows that if $E_0[T] \leq t$, then $P_0(A) \geq \frac{3}{4}$.

Let B stand for the event under which the chosen sample is smaller or equal to μ_*^*, and B^C for its complementary. Clearly, $P_0(B) = 1$.

We define the event C to be the event under which all the samples obtained from the pool are on the interval $[0, \mu_*^*]$. Clearly, $P_0(C) = 1$.

We define the event $S = \{A \cap B^C \cap C\}$, since we have already shown that $P_0(A) \geq \frac{3}{4}$, $P_0(B) = 1$ and $P_0(C) = 1$ it is obtained that $P_0(S) \geq \frac{3}{4}$. Then, since for every history of N samples, for which the event C holds, it is obtained that $\frac{dP_1}{dP_0} \geq \gamma^N$, we have the following,

$$P_1(B) \geq P_1(S) = E_0\left[\frac{dP_1}{dP_0} I(S)\right] \geq \gamma^{-4t} P_0(I(S)) \geq \frac{3}{4}\gamma^{-4t} \geq \frac{3}{4} e^{-\frac{1}{2(1-a)} \ln \frac{3}{5\delta}} \geq \delta,$$

where in the last inequality we used the facts that $(1-\epsilon)^{\frac{1}{\epsilon}} \geq e^{-1}$ and that $|supp(f_l)| \leq 1$.

We found that if a policy is (ϵ, δ)-correct under hypothesis H_0 and $E_0[T] \leq t$, then, under hypothesis H_1 this policy is not (ϵ, δ)-correct. So, for having an (ϵ, δ)-correct policy, we must have $E_0[T] > t$. Hence the lower bound is obtained. ■

7. Appendix B

Proof [Theorem 2] Since sampling from the single pool consists of choosing one pool out of the $|K|$ pools (with equal probability), and then, sampling from this pool, it follows that $f(\mu) \geq \frac{a}{|K|}$. So, $F(\mu_*^* - \epsilon) \leq \left(1 - \frac{a\epsilon}{|K|}\right)$. Also, we note that $(1-\epsilon)^{\frac{1}{\epsilon}} \leq e^{-1}$ for every $\epsilon \in (0, 1]$. Therefore, for $N = \lceil \frac{-\ln(\delta)|K|}{a\epsilon} \rceil + 1$

$$P\left(V_N^1 < \mu_*^* - \epsilon\right) = \left(F(\mu_*^* - \epsilon)\right)^N \leq \left(1 - \frac{a\epsilon}{|K|}\right)^N < \delta. \tag{11}$$

Hence, the algorithm is (ϵ, δ)-correct. The bound on the sample complexity is immediate from the definition of the algorithm. ■