Pure Exploration for Max-Quantile Bandits

Yahel David and Nahum Shimkin

Department of Electrical Engineering
Technion—Israel Institute of Technology
Haifa 32000, Israel

Abstract. We consider a variant of the pure exploration problem in Multi-Armed Bandits, where the goal is to find the arm for which the λ-quantile is maximal. Within the PAC framework, we provide a lower bound on the sample complexity of any (ϵ, δ)-correct algorithm, and propose algorithms with matching upper bounds. Our bounds sharpen existing ones by explicitly incorporating the quantile factor λ. We further provide experiments that compare the sample complexity of our algorithms with that of previous works.

1 Introduction

In the classical multi-armed bandit (MAB) problem, the learning agent faces a set K of stochastic arms, from which it chooses arms sequentially. In each round, the agent observes a random reward that depends on the selected arm. The goal of the agent is to maximize the cumulative reward (in the regret formulation), or to identify the arm with the highest expected reward (in the pure exploration problem). The MAB model has been studies extensively in the statistical and learning literature, see [2] for a comprehensive survey.

In this paper, we consider a quantile-based variant of the pure exploration MAB problem (quantile-MAB). In this variant, for a given $0 < \lambda < 1$, the goal is to identify the arm for which the λ-quantile is the largest among all arms (here, as usual the λ-quantile is such that the probability of observing a larger reward is at least λ). More precisely, considering the PAC framework, the goal is to identify an (ϵ, δ)-*correct* arm, namely an arm for which the $(\lambda - \epsilon)$-quantile is not smaller than the largest λ-quantile among all arms, with a probability larger than $1 - \delta$. In addition, we wish to minimize the sample complexity, i.e., the expected number of samples observed until the learning algorithm terminates.

For the standard MAB problem, algorithms that find the best arm (in terms of its expected reward) in the PAC sense were presented in [1, 5–8, 10], and lower bounds on the sample complexity were presented in [1, 9, 11].

Similar to the present quantile-MAB problem is the variant of the MAB problem in which the goal is to find the arm from which the largest possible sample can be obtained. This is known as the max k-armed bandit problem, and was first introduced in [3]. For this variant, algorithms that find the best arm in the PAC sense were provided in [13] and [4], and a lower bound was presented in [4]. In contrast to the current quantile-MAB problem, in the max k-armed setting, it is necessary to assume a lower bound on the tail probabilities of the arms. When the tail functions of the arms are known, and $\epsilon = \lambda$, the algorithms for the max k-armed bandit setting can be applied in the present quantile-MAB problem. However, their sample complexity upper bounds are larger than those of the algorithms presented in this paper.

More related to the present quantile-MAB problem is the work [15] which consider a measure of risk called value-at-risk (see [12]). The value-at-risk of a given random variable (R.V.) $\boldsymbol{X}$ is actually the same as the quantile of the R.V. $-\boldsymbol{X}$. An algorithm with an upper bound on the sample complexity that increases as $\frac{\lambda|K|}{\epsilon^2\delta D}$, (where D is the upper bound on the density functions) was provided in [15], that algorithm is computationally demanding since at each iteration it solves a non-linear constrained and integer-valued optimization problem. Recently, the quantile-MAB problem was studied in [14]. They provided a lower bound for the case in which $\lambda = 3/4$ and an algorithm with an upper bound on the sample complexity of the order of $\sum_{k\in K}\frac{1}{(\max(\epsilon,\Delta_{k,\lambda}))^2}\ln(\frac{|K|}{\delta\max(\epsilon,\Delta_{k,\lambda})})$, where $\Delta_{k,\lambda}$ is the difference between the λ-quantile of arm k and that of the best arm.

In this paper, for certain arm distributions, we provide a lower bound of the order of $\sum_{k\in K}\frac{\lambda(1-\lambda)\ln(\frac{1}{\delta})}{(\max(\epsilon,\Delta_{k,\lambda}))^2}$ on the sample complexity of every (ϵ,δ)-*correct* algorithm. That lower bound improves the bound in [14] in the sense of considering the quantile factor $\lambda(1-\lambda)$. This is significant when λ is close to 1 or to 0. Furthermore, for general distribution functions, we provide two algorithms that attain the lower bound up to the logarithmic terms $\ln(|K|\epsilon)$ and $\ln(|K|\log_2(\epsilon))$ respectively. The upper bounds of these algorithms are smaller than that in [14] by a factor of λ and a logarithmic factor in ϵ for the second algorithm.

The paper proceeds as follows. In the next section we present our model. In Section 3, a lower bound on the sample complexity of every (ϵ,δ)-*correct* algorithm is presented. Then in Section 4 we present our (ϵ,δ)-correct algorithms, and provide upper bounds on their sample complexity. The second algorithm is bases on applying the doubling trick on the first one. Then, in Section 5 we provide experiments that illustrate the improved sample complexity of our algorithms compared with the results presented in [14]. In Section 6 we close the paper with some concluding remarks.

2 Model Definition

We consider a finite set of arms, denoted by K. At each stage $t = 1, 2, \ldots$ the learning agent chooses an arm $k \in K$, and a real valued reward is obtained from that arm. The rewards obtained from each arm k are independent and identically distributed, with a distribution function (CDF) $F_k(x)$, $x \in \mathbb{R}$. We denote the quantile function of arm $k \in K$ by $Q_k : [0,1] \to \mathbb{R}$, and define it as follows.

Definition 1. *For every arm $k \in K$, the quantile function $Q_k(\lambda)$ is defined by*

$$Q_k(\lambda) \triangleq \inf\{x \in \mathbb{R} | 1 - \lambda < F_k(x)\}.$$

Note that $P\left(\boldsymbol{x}_k \geq Q_k(\lambda)\right) \geq 1-\lambda$ where $\boldsymbol{x}_k$ stands for a random variable with distribution F_k. Clearly, if F_k is continuous at the point $Q_k(\lambda)$, we have equality, namely, $P\left(\boldsymbol{x}_k \geq Q_k(\lambda)\right) = 1-\lambda$.

An algorithm for the quantile-MAB problem samples an arm at each time step, based on the observed history so far (i.e., the previously selected arms and observed rewards). We require the algorithm to terminate after a random number T of samples, which is finite with probability 1, and return an arm k'. An algorithm is said to be (ϵ, δ)-*correct* if the returned arm is ϵ-optimal with a probability larger than $1-\delta$, (see a precise definition later in this section). The expected number of samples $E[T]$ taken by the algorithm is the *sample complexity*, which we wish to minimize.

We next provide some definitions and notations which we use later in this paper. A λ-quantile optimal arm is defined as follows.

Definition 2. *Arm $k \in K$ is λ-quantile optimal if*

$$Q_k(\lambda) = x^*_\lambda \triangleq \max_{k' \in K} Q_{k'}(\lambda).$$

We use the following quantity which represents the distance of an arm from being optimal,

$$\varDelta_{k,\lambda} = \sup\{F_k(x) | x < x^*_\lambda\} - (1-\lambda). \tag{1}$$

If F_k is continuous, then $\varDelta_{k,\lambda} = F_k(x^*_\lambda) - (1-\lambda)$. Furthermore, note that for every suboptimal arm k, namely, an arm for which $Q_k(\lambda) < x^*_\lambda$, it follows by the monotonicity of CDF functions that $\varDelta_{k,\lambda} > 0$.

Now we are ready to precisely define an (ϵ, δ)-*correct* algorithm.

Definition 3. *For λ and ϵ such that $0 < \epsilon < \lambda < 1$ and $\delta > 0$, an algorithm is* (ϵ, δ)-correct *if*

$$P\left(Q_{k'}(\lambda - \epsilon) \geq x^*_\lambda\right) \geq 1 - \delta$$

where k' stands for the arm returned by the algorithm.

3 Lower Bound

Before presenting our algorithms, we provide a lower bound on the sample complexity of any (ϵ, δ)-*correct* algorithm for certain arm distributions. The lower bound is provided in the following Theorem.

Theorem 1. *Assume that F_k is continuous for every $k \in K$. Fix some ϵ_0 such that $0 < \epsilon_0 < \frac{1}{4}$. For every $\lambda \in [2\epsilon_0, 1-2\epsilon_0]$, $\epsilon \in (0, \epsilon_0]$ and $\delta \leq 0.15$, there exist some set of arm distributions $\{F_k\}_{k \in K}$, such that for every* (ϵ, δ)-correct *algorithm,*

$$E[T] \geq \sum_{k \in K \setminus k^*} \frac{\lambda\left(1-\lambda\right)}{2\left(\max\left(\epsilon, \varDelta_{k,\lambda}\right)\right)^2} \ln\left(\frac{1}{2.4\delta}\right) \tag{2}$$

where k^* denote some optimal arm, with $Q_{k^*}(\lambda) = x^*_\lambda$.

The above lower bound refines the one presented in [14] in the sense that here the size of the quantile λ is considered in the bound. To illustrate the lower bound, we provide an example.

Example 1. Let $\{\mu_k\}_{k\in K}$ be a set of constants, and let $\mu^* = \max_{k\in K} \mu_k$. Suppose that the rewards of each arm $k \in K$ are uniformly distributed on the interval $(\mu_k - 1, \mu_k)$. Since, $\mu_k - x^*_\lambda \leq 1$, for every arm $k \in K$ it follows that

$$\sup\{F_k(x) | x < x^*_\lambda\} = F_k(x^*_\lambda) = \begin{cases} 1 - (\mu_k - x^*_\lambda), & \mu_k \geq x^*_\lambda \\ 1, & \mu_k < x^*_\lambda \end{cases} .$$

As $x^*_\lambda = \mu^* - \lambda$, Equation (1), implies that

$$\Delta_{k,\lambda} = \min\left(\lambda, \mu^* - \mu_k\right).$$

Since $\epsilon < \lambda$, the denominator term in Equation (2) can be seen to be

$$\max\left(\epsilon, \Delta_{k,\lambda}\right) = \begin{cases} \epsilon, & \mu^* - \mu_k < \epsilon \\ \mu^* - \mu_k, & \epsilon \leq \mu^* - \mu_k < \lambda \\ \lambda, & \lambda \leq \mu^* - \mu_k \end{cases} .$$

Proof. (Theorem 1). First we assume that the quantile value of the optimal arm, namely, x^*_λ is known. Moreover, we assume that for every arm $k \in K$, the conditional probabilities $P\left(\boldsymbol{x}_k | \boldsymbol{x}_k \geq x^*_\lambda\right)$ and $P\left(\boldsymbol{x}_k | \boldsymbol{x}_k < x^*_\lambda\right)$ are also known. Therefore, the learning algorithm needs only to estimate the parameters

$$p_k \triangleq P\left(\boldsymbol{x}_k \geq x^*_\lambda\right), \quad \forall k \in K.$$

Now, by the continuity of the distribution functions it follows that $\max_{k\in K} p_k = \lambda$. Also, by Equation (1) it follows that

$$\max_{k\in K} p_k - p_{k'} = \Delta_{k',\lambda}.$$

Therefore, finding an arm k' such that $\Delta_{k',\lambda} \leq \epsilon$ is the same as finding a Bernoulli arm k', such that its expected value is ϵ-optimal, namely, $\max_{k\in K} p_k - p_{k'} \leq \epsilon$. So, our problem is the same as the standard Bernoulli bandit problem with $\{p_k\}_{k\in K}$ as the Bernoulli parameters.

Then, by Remark 5 in [9], in which a lower bound for the standard MAB problem with Bernoulli arms is provided for $\delta \leq 0.15$, we have

$$E[T] \geq \left(\frac{|S_\epsilon| - 1}{KL\left(\lambda, \lambda - \epsilon\right)} + \sum_{k\in\{K\setminus S_\epsilon\}} \frac{1}{KL\left(p_k, \lambda + \epsilon\right)} \right) \ln \frac{1}{2.4\delta}, \tag{3}$$

where $S_\epsilon \triangleq \{k | k \in K, p_k \geq \lambda - \epsilon\}$ and $KL\left(p, q\right)$ stands for the Kullback-Leibler divergence between two Bernoulli distributions with parameters p and q respectively.

We note that $\ln(1+x) \leq x$. Hence,

$$KL\left(p,q\right) = p\ln\left(\frac{p}{q}\right) + (1-p)\ln\left(\frac{1-p}{1-q}\right) \leq p\frac{p-q}{q} + (1-p)\frac{q-p}{1-q} = \frac{(p-q)^2}{q(1-q)}. \tag{4}$$

Therefore, by Equations (3) and (4) it follows that

$$E[T] \geq \left(\frac{\left(\lambda-\epsilon\right)\left(1-\lambda+\epsilon\right)\left(|S_\epsilon|-1\right)}{\epsilon^2} + \sum_{k\in\{K\setminus S_\epsilon\}} \frac{\left(\lambda+\epsilon\right)\left(1-\lambda-\epsilon\right)}{\left(\lambda+\epsilon-p_k\right)^2}\right)\ln\frac{1}{2.4\delta}.$$

Hence, by the facts that $2\epsilon \leq \lambda$ and $2\epsilon \leq (1-\lambda)$ and since $\varDelta_{k,\lambda} = \lambda - p_k$, Equation (2) is obtained. □

4 Algorithms

In this section we provide two related algorithms. The first one is simpler and attains the lower bound in Theorem 1 up to a logarithmic term. The second algorithm is based on applying the doubling trick on the first one and hence its upper bound attains Theorem 1 up to a double logarithmic term.

4.1 The Max-Q Algorithm

Here we present our Max-Q algorithm. The algorithm is (ϵ,δ)-correct and based on sampling the arm which has the highest potential λ-quantile value.

The Max-Q algorithm starts by sampling a fixed number of times from each arm. Then, for each arm, the algorithm associates a value that has been sampled from its quantile in a large probability and choses the arm for which the value is maximal. If the number of times that arm has been sampled is larger than a certain threshold, the algorithm stops returns that arm, else it samples one more time from the chosen arm.

The fundamental difference between the Max-Q algorithm and the algorithm presented in [14] is the fact that in the latter the entire CDF is estimated, while in this paper, just the value of the quantile is estimated. That difference leads to a bound on the sample complexity of the Max-Q algorithm which is smaller by a factor of λ, compared to that in [14].

Theorem 2. *For every $\lambda \in (0,1)$, $\epsilon \in (0,\lambda)$ and $\delta \in (0,1)$, Algorithm 1 is (ϵ,δ)-correct with a sample complexity bound of*

$$E[T] \leq \sum_{k\in K} \frac{10\lambda L}{\left(\max\left(\epsilon,\varDelta_{k,\lambda}\right)\right)^2} + |K| + 1, \tag{5}$$

where $L = 6\ln\left(|K|\left(1+\frac{-10\lambda\ln(\delta)}{\epsilon^2}\right)\right) - \ln\left(\delta\right)$ as defined in the algorithm.

Algorithm 1 Maximal Quantile (Max-Q) Algorithm

```
1: Input: Quantile $\lambda \in (0,1)$, constants $\delta > 0$ and $\epsilon > 0$.
   Define $L = 6\ln\left(|K|\left(1+\frac{-10\lambda\ln(\delta)}{\epsilon^2}\right)\right) - \ln(\delta)$.
2: Initialization: Counters $C(k) = N_0$, $k \in K$,
   where $N_0 = \lfloor \frac{3L}{\lambda} \rfloor + 1$.
3: Sample $N_0$ times from each arm.
4: Set $k^* \in \arg\max_{k\in K} V^k$ (with ties broken arbitrary), where $V^k$ is the $m_k$-th largest
   reward observed so far from arm $k$ and
       $m_k = \lfloor \lambda C(k) - \sqrt{3\lambda C(k) L} \rfloor + 1.$
5: if $C(k^*) > \frac{10\lambda L}{\epsilon^2}$ then
6:     Stop and return arm $k^*$.
7: else
8:     Sample once from arm $k^*$, set $C(k^*) = C(k^*) + 1$ and return to step 4.
9: end if
```

It may be observed that for $\lambda \leq \frac{1}{2}$, the upper bound provided in Theorem 2 is of the same order as the lower bound in Theorem 1, up to a logarithmic factor.

To establish Theorem 2, we first bound the probability of the event under which the m-th largest sample of one of the optimal arm is below the λ-quantile. Then, we bound the number of samples needed to be observed from each suboptimal arm such that the m-th largest value (obtained from that arm) is below the $(\lambda - \epsilon)$-quantile. For establishing these bounds in a way that the multiplicative factor of λ remains in the bounds, we use Bernstein's inequality for bounding the difference between the empirical mean and the mean value of a Bernoulli R.V. which is one if the sampled value is above the quantile and zero otherwise.

Proof. (Theorem 2). We denote the time step of the algorithm by t, the value of the counter $C(k)$ at time step t by $C^t(k)$ and we use the notations $L' = L + \ln(\delta)$ and x^* as a short for x^*_λ. Recall that T stands for the random final time step. By the condition in step 5 of the algorithm, for every arm $k \in K$, it follows that,

$$C^{T-1}(k) \leq \lfloor \frac{10\lambda\left(L' - \ln(\delta)\right)}{\epsilon^2} \rfloor + 1. \tag{6}$$

Note that by the facts that for $x \geq 6$ it follows that $\frac{d6\ln(x)}{dx} \leq 1$, and that for $x_0 = 20$ it follows that $x_0 > 6\ln(x_0) + 1$, it is obtained that

$$L'' \triangleq |K|\left(\frac{-10\lambda\ln(\delta)}{\epsilon^2} + 1\right) > 6\ln\left(|K|\left(\frac{-10\lambda\ln(\delta)}{\epsilon^2} + 1\right)\right) + 1 = L' + 1,$$

for $L'' \geq 20$. So, by the fact that $T = \sum_{k\in K} C^{T-1}(k) + 1$, for $L'' \geq 20$ it follows that

$$\begin{aligned} T &\leq |K|\left(\frac{10\lambda\left(L' - \ln(\delta)\right)}{\epsilon^2} + 1\right) + 1 < |K|\left(\frac{10\lambda\left(L'' - \ln(\delta)\right)}{\epsilon^2} + 1\right) \\ &\leq L''^2 = e^{\frac{L'}{3}}. \end{aligned} \tag{7}$$

We proceed to establish the (ϵ, δ)-correctness of the algorithm. Let $V_N^k(m)$ stand for the m-th largest value obtained from arm k after sampling it for N times and assume w.l.o.g. that $Q_1(\lambda) = x_\lambda^*$. Then, for $N \geq N_0$ and $m = \lfloor \lambda N - \sqrt{3\lambda N L} \rfloor + 1$, as stated in the algorithm, by Lemma 1 below it follows that

$$P\left(V_N^1(m) < x^*\right) \leq \delta e^{-L'}. \tag{8}$$

Hence, at every time step t, by Equations (7) and (8), applying the union bound obtains

$$P\left(V^{t,1} < x^*\right) \leq \sum_{N=N_0}^{\exp\left(\frac{L'}{3}\right)} P\left(V_N^1(m) < x^*\right) \quad = \delta e^{-\frac{2L'}{3}}. \tag{9}$$

where $V^{t,k}$ stands for the value of V^k at time step t.

Let k_T^* stand for the arm returned by the algorithm. Also, by Lemma 1, for $N > \frac{10\lambda\left(L' - \ln(\delta)\right)}{\epsilon^2}$, it follows that

$$P\left(V_N^k(m) > Q_k(\lambda - \epsilon)\right) \leq \delta e^{-L'}. \tag{10}$$

So, since by the condition in step 5, it is obtained that $C(k_T^*) > \frac{10\lambda\left(L' - \ln(\delta)\right)}{\epsilon^2}$, it follows by Equation (10) and the union bound that

$$P\left(V^{T,k_T^*} > Q_{k_T^*}(\lambda - \epsilon)\right) \leq \sum_{k \in K} \sum_{t=1}^{\exp\left(\frac{L'}{3}\right)} \sum_{N=1}^{\exp\left(\frac{L'}{3}\right)} \delta e^{-L'} = |K| \delta e^{-\frac{L'}{3}}. \tag{11}$$

Also, by Equation (9) and the union bound it follows that

$$P\left(V^{T,1} < x^*\right) \leq \sum_{t=1}^{\exp\left(\frac{L'}{3}\right)} P\left(V^{t,1} < x^*\right) \leq \delta e^{-\frac{L'}{3}}. \tag{12}$$

So, since by step 4 of the algorithm, $V^{T,k_T^*} \geq V^{T,1}$, it follows by Equations (11) and (12) that

$$P\left(Q_{k_T^*}(\lambda - \epsilon) < x^*\right) \leq P\left(V^{T,k_T^*} > Q_{k_T^*}(\lambda - \epsilon)\right) + P\left(V^{T,1} < x^*\right) < \delta.$$

It follows that the algorithm returns an ϵ-optimal arm with a probability larger than $1 - \delta$. Hence, it is (ϵ, δ)-correct.

To prove the bound on the expected sample complexity of the algorithm, we define the following sets:

$$M(\epsilon) = \{l \in K | \Delta_{k,\lambda} \leq \epsilon\} \quad \text{and} \quad N(\epsilon) = \{l \in K | \Delta_{k,\lambda} > \epsilon\}.$$

As before, we assume w.l.o.g. that $Q_1(\lambda) = x^*$. Then, for the case in which

$$E_1 \triangleq \bigcap_{1 \le t \le T} \left\{ V^{t,1} \ge x^* \right\}$$

occurs, for every arm $k \in K$, a necessary condition for $C^T(k) > N'_k$, where $N'_k = \lfloor \frac{10\lambda\left(L' - \ln(\delta)\right)}{\Delta^2_{k,\lambda}} \rfloor + 1$ is

$$E_k \triangleq \left\{ V^k_{N'_k}(m'_k) \ge x^* \right\},$$

where $m'_k = \lfloor \lambda N'_k - \sqrt{3\lambda N'_k \left(L' - \ln(\delta)\right)} \rfloor + 1$.

Now, by using the bound in Equation (6) and the fact that $\sum_{k \in K} C^T(k) = \sum_{k \in K} C^{T-1}(k) + 1$ for the arms in the set $M(\epsilon)$, N'_k as a bound for the arms in the set $N(\epsilon)$, and the bound in Equation (7), it is obtained that

$$\begin{aligned} E[T] \le (1 - P(E_1))\, e^{\frac{L'}{3}} + P(E_1) \sum_{k \in N(\epsilon)} \Big((1 - P(E_k|E_1))\, \varPhi_k(\epsilon) + e^{\frac{L'}{3}} P(E_k|E_1) \Big) \\ + \sum_{k \in M(\epsilon)} \varPhi_k(\epsilon) + 1, \end{aligned} \tag{13}$$

where $\varPhi_k(\epsilon) = \lfloor \frac{10\lambda\left(L' - \ln(\delta)\right)}{(\max(\epsilon, \Delta_{k,\lambda}))^2} \rfloor + 1$. But, by Equation (9) it follows that

$$P(E_1) \ge 1 - \sum_{t=1}^{\exp\left(\frac{L'}{3}\right)} P\left(V^{t,1} < x^*\right) \ge 1 - \delta e^{\frac{-2L'}{3}} e^{\frac{L'}{3}} = 1 - \delta e^{\frac{-L'}{3}}. \tag{14}$$

Also, since $Q'_k \triangleq Q_k\left(\lambda - \sqrt{\frac{10\lambda(L' - \ln(\delta))}{N'_k}}\right) < x^*$ for $k \in N(\epsilon)$, it follows by Lemma 1 that

$$P(E_k|E_1)\, P(E_1) \le P(E_k) \le P\left(V^k_{N'_k}(m'_k) > Q'_k\right) \le \delta e^{-L'}, \quad \forall k \in N(\epsilon) \tag{15}$$

Therefore, by Equations (13), (14) and (15) and the definition of $\varPhi_k(\epsilon)$, the bound on the sample complexity is obtained. □

Lemma 1. *For every arm $k \in K$, let $V^k_N(m)$ stand for the m-th largest value obtained from arm k after sampling it for N times. Then, for any positive integers m and N such that $m < N$, and every $\lambda \in [0, 1]$, it follows that,*

1. *If $\frac{m}{N} > \lambda$, then*
$$P\left(V^k_N(m) > Q_k(\lambda)\right) \le f_0(m, N, \lambda).$$
2. *If $\frac{m}{N} < \lambda$, then*
$$P\left(V^k_N(m) < Q_k(\lambda)\right) \le f_0(m, N, \lambda),$$
where $f_0(m, N, \lambda) = \exp\left(-\frac{|m - N\lambda|^2}{2(N\lambda + |m - N\lambda|/3)}\right)$.

The proof is based on Bernstein's inequality.

Proof. In this proof, we omit the arm index k for short. We start with claim (1). Let x_i stand for the i-th sampled value from the arm, and let $\{X_i(\lambda)\}$ and $\{Y_i(\lambda)\}$ be random variables for which

$$X_i(\lambda) = \begin{cases} 1 & w.p \quad \lambda \\ 0 & w.p \quad 1-\lambda \end{cases} \quad \text{and} \quad Y_i(\lambda) \quad = \begin{cases} 1 & x_i > Q(\lambda) \\ 0 & x_i \leq Q(\lambda) \end{cases} . \tag{16}$$

Note that the variables $\{Y_i(\lambda)\}$ are i.i.d. The variables $\{X_i(\lambda)\}$ are i.i.d as well. Then, since $P\left(Y_i(\lambda)=1\right) \leq P\left(X_i(\lambda)=1\right)$, after sampling N times,

$$\begin{aligned} P\left(V_N(m) > Q(\lambda)\right) &= P\left(\frac{1}{N}\sum_{i=1}^{N} Y_i(\lambda) \geq \frac{m}{N}\right) \leq P\left(\frac{1}{N}\sum_{i=1}^{N} X_i(\lambda) \geq \frac{m}{N}\right) \\ &= P\left(\frac{1}{N}\sum_{i=1}^{N} \widetilde{X}_i(\lambda) \geq \frac{m}{N} - E[X_1(\lambda)]\right) \triangleq \varUpsilon(\lambda, m, N)\,, \end{aligned} \tag{17}$$

where $\widetilde{X}_i(\lambda) = X_i(\lambda) - E[X_1(\lambda)]$. So, $\left\{\widetilde{X}_i(\lambda)\right\}$ satisfies the conditions of Bernstein's inequality with $\sigma^2 = \lambda\left(1-\lambda\right)$, and $E[X_1(\lambda)] = \lambda$. Therefore

$$\begin{aligned} \varUpsilon(\lambda, m, N) &\leq \exp\left(-\frac{\left(m - N\lambda\right)^2}{2\left(N\lambda\left(1-\lambda\right) + \left(m - N\lambda\right)/3\right)}\right) \\ &\leq \exp\left(-\frac{\left(m - N\lambda\right)^2}{2\left(N\lambda + \left(m - N\lambda\right)/3\right)}\right) . \end{aligned} \tag{18}$$

Proceeding to claim (2), let $\{Z_i(\lambda)\}$ be random variables for which

$$Z_i(\lambda) = \begin{cases} 1 & x_i \geq Q(\lambda) \\ 0 & x_i < Q(\lambda) \end{cases} . \tag{19}$$

Note that $\{Z_i\}$ are i.i.d. Then, since $P\left(Z_i(\lambda)=1\right) \geq P\left(X_i(\lambda)=1\right)$,

$$\begin{aligned} P\left(V_N(m) < Q(\lambda)\right) &= P\left(\frac{1}{N}\sum_{i=1}^{N} Z_i(\lambda) < \frac{m}{N}\right) \leq P\left(\frac{1}{N}\sum_{i=1}^{N} X_i(\lambda) \leq \frac{m}{N}\right) \\ &= P\left(\frac{1}{N}\sum_{i=1}^{N} \widetilde{X}_i(\lambda) \leq \frac{m}{N} - E[X_1(\lambda)]\right) \triangleq \hat{\varUpsilon}(\lambda, m, N) \end{aligned} \tag{20}$$

and by symmetry

$$\begin{aligned} \hat{\varUpsilon}(\lambda, m, N) &\leq \exp\left(-\frac{\left(N\lambda - m\right)^2}{2\left(N\lambda\left(1-\lambda\right) + \left(N\lambda - m\right)/3\right)}\right) \\ &\leq \exp\left(-\frac{\left(N\lambda - m\right)^2}{2\left(N\lambda + \left(N\lambda - m\right)/3\right)}\right) . \end{aligned} \tag{21}$$

□

Algorithm 2 Doubled Maximal Quantile (Max-Q) Algorithm

```
1: Input: Quantile $\lambda \in (0,1)$, constants $\delta > 0$ and $\epsilon > 0$.
   Define $L_D = 6\ln\left(|K|\log_2\left(\frac{-20\lambda\ln(\delta)}{\epsilon^2}\right)\right) - \ln(\delta)$.
2: Initialization: Counters $C(k) = N_0$, $k \in K$,
   where $N_0 = \lfloor \frac{3L_D}{\lambda} \rfloor + 1$.
3: Sample $N_0$ times from each arm.
4: Set $k^* \in V^k$ (with ties broken arbitrary), where $V^k$ is the $m_k$-th largest reward
   observed so far from arm $k$ and
       $$m_k = \lfloor \lambda C(k) - \sqrt{3\lambda C(k) L_D} \rfloor + 1.$$
5: if $C(k^*) > \frac{10\lambda L_D}{\epsilon^2}$ then
6:     Stop and return arm $k^*$.
7: else
8:     Sample $C(k^*)$ times from arm $k^*$, set $C(k^*) = 2C(k^*)$ and return to step 4.
9: end if
```

4.2 The Doubled Max-Q Algorithm

Here we improve on the previous algorithm by resorting to the doubling trick. The Doubled Max-Q Algorithm is based on the same principle as the Max-Q Algorithm. However, instead of observing one sample at each time step, here the algorithm doubles the number of samples of the chosen arm. Consequently, the number of times at which the algorithm needs to choose an arm is roughly logarithmic compared to that under the previous algorithm, leading to a tighter bound. Algorithm 2 presents the proposed Doubled Max-Q algorithm.

Theorem 3. *For every $\lambda \in (0,1)$, $\epsilon \in (0,\lambda)$ and $\delta \in (0,1)$, Algorithm 2 is (ϵ,δ)-correct with a sample complexity bound of*

$$E[T] \leq \sum_{k\in K} \frac{20\lambda L_D}{\left(\max\left(\epsilon, \Delta_{k,\lambda}\right)\right)^2} + |K| + 1, \tag{22}$$

where $L_D = 6\ln\left(|K|\log_2\left(\frac{-20\lambda\ln(\delta)}{\epsilon^2}\right)\right) - \ln(\delta)$ as defined in the algorithm.

Here, the upper bound is of the same order as the lower bound in Theorem 1, up to a double-logarithmic order.

The proof of Theorem 3 is established by some adjustments of the proof of Theorem 2.

Proof. As before, we denote the time step of the algorithm by t, the value of the counter $C(k)$ at time step t by $C^t(k)$ and we use the notations $L'_D = L_D + \ln(\delta)$ and x^* as a short for x^*_λ. We note that here, at each time step, there may be more than a single sample, so T, the sample complexity, may be different than the final time step. Hence, here we denote the (random) final time step by T_D. By the condition in step 5 of the algorithm, for every arm $k \in K$, it follows that,

$$C^{T_D-1}(k) \leq \frac{10\lambda\left(L'_D - \ln(\delta)\right)}{\epsilon^2}. \tag{23}$$

Note that by the facts that for $x \geq 6$ it follows that $\frac{d6\ln(x)}{dx} \leq 1$, and that for $x_0 = 20$ it follows that $x_0 > 6\ln(x_0) + 1$ it is obtained that

$$L''_D \triangleq |K|\log_2\left(\frac{-20\lambda\ln(\delta)}{\epsilon^2}\right) \quad > 6\ln\left(|K|\log_2\left(\frac{-20\lambda\ln(\delta)}{\epsilon^2}\right)\right) + 1 = L'_D + 1,$$

for $L''_D \geq 20$. So, by the fact that $T = \sum_{k\in K}\log_2\left(2C^{T_D-1}(k)\right)$, for $L''_D \geq 20$ it follows that

$$\begin{aligned} T &\leq |K|\log_2\left(\frac{20\lambda\left(L'_D - \ln(\delta)\right)}{\epsilon^2}\right) < |K|\log_2\left(\frac{20\lambda\left(L''_D - \ln(\delta)\right)}{\epsilon^2}\right) \\ &\leq |K|\log_2\left(L''_D\right) + L''_D \leq L''_D\left(\log_2\left(L''_D\right) + 1\right) \leq \frac{\left(L''_D\right)^2}{2} = \frac{1}{2}e^{\frac{L'_D}{3}}. \end{aligned} \tag{24}$$

Recall that x^* is used as a short for x^*_λ. Now, we begin with proving the (ϵ, δ)-correctness property of the algorithm. We let $V^k_N(m)$ stands for the m-th largest value obtained from arm k after sampling it for N times and we assume w.l.o.g. that $Q_1(\lambda) = x^*$. Then, for $N \geq N_0$ and $m = \lfloor \lambda N - \sqrt{3\lambda N L_D} \rfloor + 1$, as stated in the algorithm, by Lemma 1 it follows that

$$P\left(V^1_N(m) < x^*\right) \leq \delta e^{-L'_D} \tag{25}$$

Hence, at every time step t, by Equations (24) and (25), by applying the union bound, for $N_i = 2^i N_0$ it follows that

$$P\left(V^{t,1} < x^*\right) \leq \sum_{i=0}^{\frac{1}{2}\exp\left(\frac{L'_D}{3}\right)} P\left(V^1_{N_i}(m) < x^*\right) \quad = \delta e^{-\frac{2L'_D}{3}}. \tag{26}$$

where $V^{t,k}$ stands for the value of V^k at time step t.

Now, we let $k^*_{T_D}$ stands for the arm returned by the algorithm. Also, by Lemma 1, for $N > \frac{10\lambda\left(L'_D - \ln(\delta)\right)}{\epsilon^2}$, it follows that

$$P\left(V^k_N(m) > Q_k(\lambda - \epsilon)\right) \leq \delta e^{-L'_D}. \tag{27}$$

So, since by the condition in step 5, it is obtained that $C(k^*_{T_D}) > \frac{10\lambda\left(L'_D - \ln(\delta)\right)}{\epsilon^2}$, it follows by Equation (27) and the union bound that

$$P\left(V^{T_D,k^*_{T_D}} > Q_{k^*_{T_D}}(\lambda - \epsilon)\right) \leq \sum_{k\in K}\sum_{t=1}^{\frac{1}{2}\exp\left(\frac{L'_D}{3}\right)}\sum_{i=0}^{\frac{1}{2}\exp\left(\frac{L'_D}{3}\right)} \delta e^{-L'_D} = |K|\delta e^{-\frac{L'_D}{3}}. \tag{28}$$

Also, by Equation (26) and applying the union bound it follows that

$$P\left(V^{T_D,1} < x^*\right) \leq \sum_{t=1}^{\frac{1}{2}\exp\left(\frac{L'_D}{3}\right)} P\left(V^{t,1} < x^*\right) \leq \delta e^{-\frac{L'_D}{3}} \tag{29}$$

So, since by step 4 of the algorithm, $V^{T_D,k^*_{T_D}} \geq V^{T_D,1}$, it follows by Equations (28) and (29) that

$$P\left(Q_{k^*_{T_D}}(\lambda-\epsilon) < x^*\right) \leq P\left(V^{T_D,k^*_{T_D}} > Q_{k^*_{T_D}}(\lambda-\epsilon)\right) + P\left(V^{T_D,1} < x^*\right) < \delta$$

Therefore, it follows that the algorithm returns an ϵ-optimal arm 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:

$$M(\epsilon) = \{l \in K | \Delta_{k,\lambda} \leq \epsilon\} \quad \text{and} \quad N(\epsilon) = \{l \in K | \Delta_{k,\lambda} > \epsilon\}.$$

As before, we assume w.l.o.g. that $Q_1(\lambda) = x^*$. For the case in which

$$E_1 \triangleq \bigcap_{1\leq t\leq T} \left\{V^{t,1} \geq x^*\right\},$$

occurs, for every arm $k \in K$, a necessary condition for $C^{T_D}(k) > N'_{k,D}$, where

$$N'_{k,D} \triangleq \min\left\{N_i | N_i > \frac{10\lambda\left(L'_D - \ln(\delta)\right)}{\Delta^2_{k,\lambda}}, i \in \mathbb{N}\right\}$$

is

$$E_{k,D} \triangleq \left\{V^k_{N'_{k,D}}(m'_{k,D}) \geq x^*\right\},$$

where $m'_{k,D} = \lfloor \lambda N'_{k,D} - \sqrt{3\lambda N'_{k,D}\left(L'_D - \ln(\delta)\right)}\rfloor + 1$.

Then for

$$\Phi_{k,D}(\epsilon) = \frac{20\lambda\left(L'_D - \ln(\delta)\right)}{\left(\max\left(\epsilon, \Delta_{k,\lambda}\right)\right)^2}$$

for $k \in N(\epsilon)$ it follows that

$$N'_{k,D} \leq \Phi_{k,D}(\epsilon)$$

So, by using the bound in Equation (23) and the fact that $\sum_{k\in K} C^{T_D}(k) = 2\sum_{k\in K} C^{T_D-1}(k)$ for the arms in the set $M(\epsilon)$, $N'_{k,D}$ as a bound for the arms in the set $N(\epsilon)$ and the bound in Equation (24), it is obtained that

$$\begin{aligned} E[T] \leq{}& \left(1 - P\left(E_1\right)\right) e^{\frac{L'_D}{3}} \\ &+ P\left(E_1\right) \sum_{k\in N(\epsilon)} \left(\left(1 - P\left(E_{k,D}|E_1\right)\right)\Phi_{k,D}(\epsilon) + e^{\frac{L'_D}{3}} P\left(E_{k,D}|E_1\right)\right) \\ &+ \sum_{k\in M(\epsilon)} \Phi_{k,D}(\epsilon) + 1, \end{aligned} \tag{30}$$

But, by Equation (26) it follows that

$$P(E_1) \geq 1 - \sum_{t=1}^{\exp\left(\frac{L'_D}{3}\right)} P\left(V^{t,1} < x^*\right) \geq 1 - \delta e^{-\frac{2L'_D}{3}} e^{\frac{L'_D}{3}} = 1 - \delta e^{\frac{-L'_D}{3}} \qquad (31)$$

Also, since $Q_k\left(\lambda - \sqrt{\frac{10\lambda\left(L'_D - \ln(\delta)\right)}{N'_{k,D}}}\right) < x^*$ for $k \in N(\epsilon)$, it follows by Lemma 1 that

$$P(E_{k,D}|E_1)\, P(E_1) < \delta e^{-L'_D}, \quad \forall k \in N(\epsilon) \qquad (32)$$

Therefore, by Equations (30), (31) and (32) and the definition of $\Phi_{k,D}(\epsilon)$, the bound on the sample complexity is obtained. □

5 Experiments

In this section we investigate numerically the Max-Q and the Double-Max-Q algorithms presented in this paper and compare them with the QPAC algorithm presented in [14].

In Figure 1, we present the average sample complexity of 10 runs vs. the quantile λ for $\delta = 0.01$ and various values of ϵ. As shown in Figure 1, and detailed in Tables 1-4, the Max-Q and the Double-Max-Q algorithms significantly outperform the QPAC algorithm. The arms distribution functions used here were uniform with an interval of length 1.

$\log_{10}(E[T])$	Algorithm		
	QPAC	Double-Max-Q	Max-Q
	$\epsilon = 0.005$	$\epsilon = 0.005$	$\epsilon = 0.005$
$1 - \lambda = 0.8$	7.26	7.43	7.6
$1 - \lambda = 0.85$	7.26	6.47	6.75
$1 - \lambda = 0.9$	7.26	6.35	6.57
$1 - \lambda = 0.95$	7.26	6.05	6.25
$1 - \lambda = 0.96$	7.26	5.85	6.16
$1 - \lambda = 0.97$	7.26	5.68	6.04
$1 - \lambda = 0.98$	7.26	5.58	5.88
$1 - \lambda = 0.99$	7.28	5.42	5.72

Table 1. $\log_{10}$ of the average sample complexity of the Max-Q, the Double-Max-Q and the QPAC algorithms. The number of arms was 10 and the averages were computed over 10 runs.

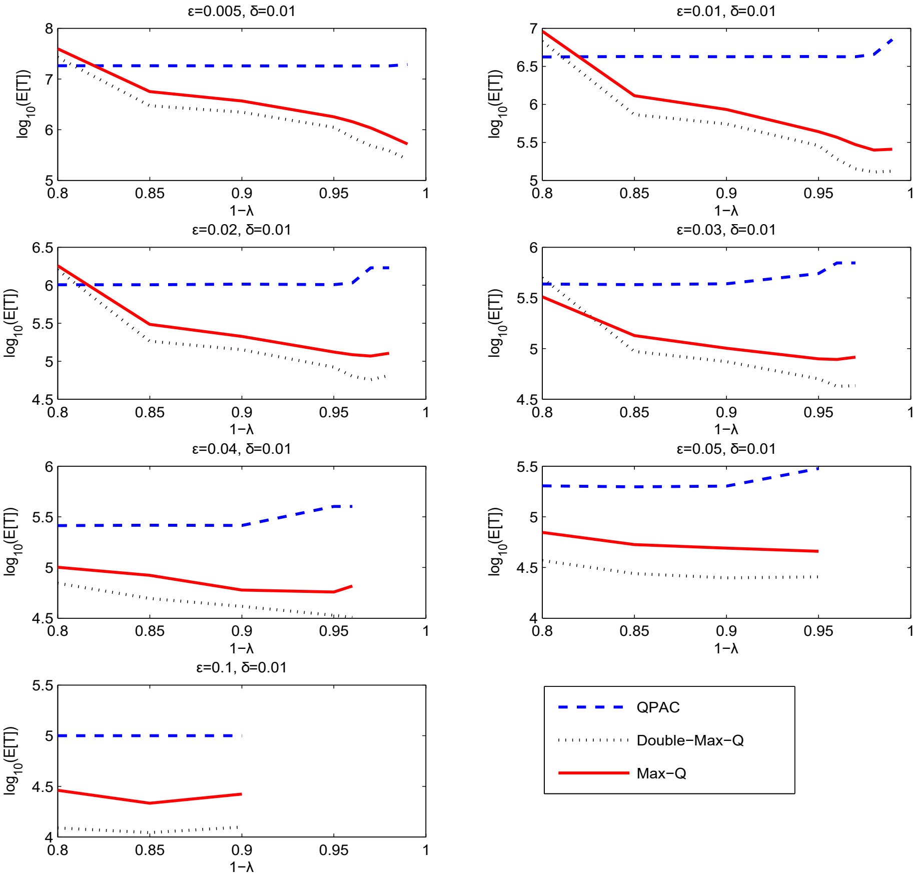

Fig. 1. The average sample complexity of the Max-Q, the Double-Max-Q and the QPAC algorithms for various of parameters settings. The number of arms was 10 and the averages were computed over 10 runs.

$\log_{10}(E[T])$	Algorithm					
	QPAC		Double-Max-Q		Max-Q	
	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.01$	$\epsilon = 0.02$	$\epsilon = 0.01$	$\epsilon = 0.02$
$1 - \lambda = 0.8$	6.22	6.01	6.83	6.2	6.96	6.26
$1 - \lambda = 0.85$	6.63	6.01	5.86	5.26	6.11	5.48
$1 - \lambda = 0.9$	6.63	6.01	5.74	5.15	5.93	5.33
$1 - \lambda = 0.95$	6.63	6.01	5.46	4.92	5.64	5.12
$1 - \lambda = 0.96$	6.63	6.03	5.28	4.8	5.57	5.09
$1 - \lambda = 0.97$	6.63	6.23	5.15	4.76	5.47	5.07
$1 - \lambda = 0.98$	6.66	6.23	5.11	4.81	5.4	5.1
$1 - \lambda = 0.99$	6.85	—	5.12	—	5.41	—

Table 2. $\log_{10}$ of the average sample complexity of the Max-Q, the Double-Max-Q and the QPAC algorithms. The number of arms was 10 and the averages were computed over 10 runs.

$\log_{10}(E[T])$	Algorithm					
	QPAC		Double-Max-Q		Max-Q	
	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.03$	$\epsilon = 0.04$	$\epsilon = 0.03$	$\epsilon = 0.04$
$1 - \lambda = 0.8$	5.64	5.41	5.64	4.85	5.51	5
$1 - \lambda = 0.85$	5.63	5.42	5.63	4.69	5.13	4.92
$1 - \lambda = 0.9$	5.64	5.41	5.64	4.62	5	4.78
$1 - \lambda = 0.95$	5.74	5.6	5.74	4.53	4.9	4.76
$1 - \lambda = 0.96$	5.85	5.6	5.85	4.51	4.89	4.82
$1 - \lambda = 0.97$	5.85	—	5.85	—	4.92	—

Table 3. $\log_{10}$ of the average sample complexity of the Max-Q, the Double-Max-Q and the QPAC algorithms. The number of arms was 10 and the averages were computed over 10 runs.

$\log_{10}(E[T])$	Algorithm					
	QPAC		Double-Max-Q		Max-Q	
	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.1$
$1 - \lambda = 0.8$	5.31	5	4.57	4.09	4.85	4.46
$1 - \lambda = 0.85$	5.3	5	4.44	4.04	4.73	4.33
$1 - \lambda = 0.9$	5.3	5	4.4	4.1	4.69	4.42
$1 - \lambda = 0.95$	5.48	—	4.41	—	4.66	—

Table 4. $\log_{10}$ of the average sample complexity of the Max-Q, the Double-Max-Q and the QPAC algorithms. The number of arms was 10 and the averages were computed over 10 runs.

6 Conclusion

In this paper we studied the pure exploration problem where the goal is to find the arm with the maximal λ-quantile. Under the PAC framework, we provided a lower bound and algorithms that attain it up to a logarithmic term (for the first algorithm) and a double-logarithmic term (for the second algorithm).

A challenge for future work is closing the logarithmic gap between the lower and upper bounds.

References

1. J.-Y. Audibert and S. Bubeck. Best arm identification in multi-armed bandits. In *Proceeding of the 23rd Conference on Learning Theory (COLT)*, pages 41–53, 2010.
2. S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
3. V. A. Cicirello and S. F. Smith. The max k-armed bandit: A new model of exploration applied to search heuristic selection. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pages 1355–1361, 2005.

4. Y. David and N. Shimkin. PAC lower bounds and efficient algorithms for the max k-armed bandit problem. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 878–887, 2016.
5. E. Even-Dar, S. Mannor, and Y. Mansour. PAC bounds for multi-armed bandit and markov decision processes. In *Proceedings of the 15th Conference on Learning Theory (COLT)*, pages 255–270. 2002.
6. V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems 25*, pages 3212–3220. Curran Associates, Inc., 2012.
7. S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 655–662, 2012.
8. Z. S. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1238–1246, 2013.
9. E. Kaufmann, O. Cappé, and A. Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.
10. E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceeding of the 26th Conference on Learning Theory (COLT)*, pages 228–251, 2013.
11. S. Mannor and J. N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:623–648, 2004.
12. B. Schachter. An irreverent guide to value at risk. *Financial Engineering News*, 1(1):17–8, 1997.
13. M. J. Streeter and S. F. Smith. An asymptotically optimal algorithm for the max k-armed bandit problem. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, pages 135–142, 2006.
14. B. Szörényi, R. Busa-Fekete, P. Weng, and E. Hüllermeier. Qualitative multi-armed bandits: A quantile-based approach. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1660–1668, 2015.
15. J. Y. Yu and E. Nikolova. Sample complexity of risk-averse bandit-arm selection. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.