Quantile-Regret Minimisation in Infinitely Many-Armed Bandits

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan
Department of Computer Science and Engineering
Indian Institute of Technology Bombay, Mumbai 400076, India
{arghya, shivaram}@cse.iitb.ac.in

Abstract

The stochastic multi-armed bandit is a well-studied abstraction of decision making in the face of uncertainty. We consider the setting in which the number of bandit arms is much larger than the possible number of pulls, and can even be infinite. With the aim of minimising regret with respect to an *optimal* arm, existing methods for this setting either assume some structure over the set of arms (Kleinberg et al., 2008, Ray Chowdhury and Gopalan, 2017), or some property of the reward distribution (Wang et al., 2008). Invariably, the validity of such assumptions—and therefore the performance of the corresponding methods—depends on instance-specific parameters, which might not be known beforehand.

We propose a conceptually simple, parameter-free, and practically effective alternative. Specifically we introduce a notion of regret with respect to the top *quantile* of a probability distribution over the expected reward of randomly drawn arms. Our main contribution is an algorithm that achieves sublinear "quantile-regret", both (1) when it is specified a quantile, and (2) when the quantile can be any (unknown) positive value. The algorithm needs no side information about the arms or about the structure of their reward distributions: it relies on random sampling to reach arms in the top quantile. Experiments show that our algorithm outperforms several previous methods (in terms of conventional regret) when the latter are not tuned well, and often even when they are.

1 INTRODUCTION

The stochastic multi-armed bandit (Berry and Fristedt, 1985) is a well-studied abstraction of on-line learning. Each bandit *arm* represents a slot-machine with a fixed (but unknown) real-valued reward distribution. An experimenter is allowed to *pull* an arm at every time instant and observe its reward. The experimenter aims to maximise the total expected reward obtained over a horizon, or equivalently, to minimise the *regret* with respect to a strategy that always plays an optimal arm. Side information regarding the bandit instance may or may not be available to the experimenter.

Regret minimisation algorithms have to achieve a balance between exploration (gathering information about the reward distributions of arms) and exploitation (pulling seemingly-good arms). For a K-armed bandit, the optimal regret that can be achieved after T pulls is $\Omega(\sqrt{KT})$ (Auer et al., 2003). To achieve a regret of $O(\sqrt{KT})$ (Audibert and Bubeck, 2009), algorithms invariably have to maintain separate statistics for the pulls coming from each arm (since any of them could be the sole optimal arm).

In many modern applications of bandits, the set of arms that can be pulled is very large, often even infinite. Examples include (1) sensor networks, in which a central controller must learn to deploy the most accurate sensor from among a large number of noisy sensors (Kadono and Fukuta, 2014); (2) crowd-sourcing tasks, in which a periodic task should ideally be assigned to the most skilled worker in a large pool (Tran-Thanh et al., 2014); (3) on-line advertising, in which a layout for an ad should be chosen from among a large set so as to maximise the click-through rate (Tang et al., 2013). Clearly, the $\Theta(\sqrt{KT})$ bound on the regret is not helpful when $K \gg T$ or $K = \infty$. Perhaps the most common apprach to deal with infinitely-many armed bandits is to utilise some sort of side information about the arms: for example, to assume that the arms are embedded in a metric space in which the reward function is Lipschitz continuous (Kleinberg, 2005) or even linear (Auer, 2003, Chu et al., 2011). Unfortunately such side information is not always available; even if available, it is often not accurate (Ghosh et al., 2017).

In this paper, we propose an approach for exploring the arms of infinitely many-armed bandits with no recourse to side information. Naturally, we cannot always guarantee sublinear regret with respect to optimal arms (which may never get pulled in any finite horizon). Instead, assuming that the

set of arms $\mathcal{A}$ is being accessed through a given sampling distribution $P_{\mathcal{A}}$, we benchmark regret against *quantiles* of the reward distribution induced by $P_{\mathcal{A}}$. By fixing the quantile, $P_{\mathcal{A}}$ will eventually sample "good enough" arms. Using a doubling trick, we can also ensure that our algorithm will eventually sample every arm whose expected reward is bounded away from the optimal reward. Below we formalise the notion of *quantile regret* and outline our contributions.

Problem Definition. A *bandit instance* $\mathcal{B} = (\mathcal{A}, M)$ comprises a (possibly infinite) set of *arms* $\mathcal{A}$, and a reward function M that gives a bounded, real-valued, reward distribution $M(a)$ for each arm $a \in \mathcal{A}$. When pulled, arm a produces a reward drawn i.i.d. from $M(a)$. We denote the expected reward from arm a as $\mu_a \stackrel{\text{def}}{=} \mathbb{E}[M(a)]$. Without loss of generality, we assume that all rewards lie in $[0, 1]$.

An *algorithm* is a mapping from the set of histories (of arms pulled and rewards obtained) to the set of probability distributions over $\mathcal{A}$: thus, given a history, an algorithm specifies a probability for sampling each arm. Let $\mu^* \stackrel{\text{def}}{=} \min\{y \in [0, 1] : \forall a \in \mathcal{A},\ \mu_a \leq y\}$. Then, for a given horizon of pulls T, the *regret* of an algorithm is

$$\mathbb{R}^*_T = T\mu^* - \sum_{t=1}^{T} \mathbb{E}[\mu_{a_t}], \tag{1}$$

where a_t denotes the arm pulled by the algorithm at time t. The expectation is taken over the random outcomes of pulls, as well as random choices (if any) made by the algorithm.

For us, a *problem instance* $\mathcal{I} = (\mathcal{B}, P_{\mathcal{A}})$ contains a bandit instance $\mathcal{B}$ with arms $\mathcal{A}$, and a sampling distribution $P_{\mathcal{A}}$ to choose arms from $\mathcal{A}$. We apply the concept of "(ϵ, ρ)-optimality" introduced by Roy Chaudhuri and Kalyanakrishnan (2017). For a *quantile fraction* $\rho \in [0, 1]$ and a *tolerance* $\epsilon \in [0, 1]$, an arm $a \in \mathcal{A}$ is said to be (ϵ, ρ)-optimal if $\Pr_{a' \sim_{P_{\mathcal{A}}}}\{\mu_a \geq \mu_{a'} - \epsilon\} \geq 1 - \rho$. Let $\mathcal{TOP}_\rho$ be the set of all $(0, \rho)$-optimal arms. Also let $\mu_\rho \in [0, 1]$ be a quantity such that, $\forall a' \in \mathcal{A} \setminus \mathcal{TOP}_\rho,\ \mu_\rho > \mu_{a'}$ and $\forall a \in \mathcal{TOP}_\rho,\ \mu_a \geq \mu_\rho$. In other words, if μ denotes the mean of an arm drawn according to $P_{\mathcal{A}}$, then μ_ρ is the $(1 - \rho)$-th quantile of the distribution of μ.

Figure 1 shows the example of an infinite set of arms $\mathcal{A}$ whose means lie between 0.15 and 0.95. When sampled according to $P^1_{\mathcal{A}}$, the resulting probability density function of the mean μ is $D^1(\mu)$. If an algorithm is constrained to access arms using $P^1_{\mathcal{A}}$, it is only natural for us to evaluate it against a baseline that is also constrained by $P^1_{\mathcal{A}}$. For example, we could aim for the algorithm to eventually surpass q^1, which is the 94-th percentile of the distribution induced by $P^1_{\mathcal{A}}$. Without additional information, such an algorithm cannot hope to surpass q^2, the 94-th percentile of a different sampling distribution $P^2_{\mathcal{A}}$. In general we expect that there is some "natural" way to sample the unknown set of arms—and this is given to us as $P_{\mathcal{A}}$. For example, if $\mathcal{A}$ is finite, it is a reasonable choice to assign an equal probability to

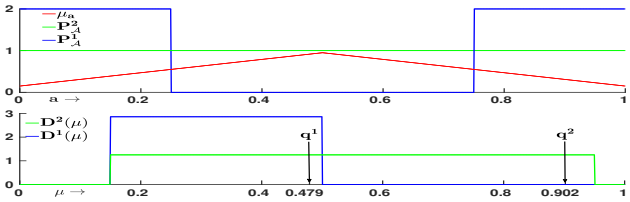

Figure 1: Two problem instances that share the same set of arms $\mathcal{A}$, varying continuously in $[0, 1]$. The top panel shows the expected rewards μ_a of each arm $a \in \mathcal{A}$, as well as probability density functions $P^1_{\mathcal{A}}$ and $P^2_{\mathcal{A}}$ for sampling $\mathcal{A}$. The bottom panel shows the reward distributions $D^1_{\mathcal{A}}$ and $D^2_{\mathcal{A}}$ induced by each of the sampling distributions, along with 94-th percentiles q^1 and q^2, respectively.

sampling each arm. If $\mathcal{A}$ corresponds to a parameterised set, $P_{\mathcal{A}}$ could implement some distribution over the parameters.

We are now ready to define *quantile-regret* with respect to a given quantile fraction ρ. If a_t is the arm drawn by an algorithm in its t^{th} pull, we define the (cumulative) quantile-regret (or the "ρ-regret") after T pulls as

$$\mathbb{R}_T(\rho) = T\mu_\rho - \sum_{t=1}^{T} \mathbb{E}[\mu_{a_t}]. \tag{2}$$

Our aim is to devise algorithms to minimise $\mathbb{R}_T(\rho)$ for a given problem instance $\mathcal{I}$. The quantile fraction ρ can either be given as an input to the algorithm, or left unspecified, as we describe next.

Contributions. We show that without any knowledge of the reward distributions of the arms, or of side information such as a distance metric over arms, it is still possible to have strategies to maximise reward over time. We articulate the problem as that of achieving sub-linear ρ-regret.

1. In Section 3.1, we present our first algorithm, QRM1, which is given a quantile fraction ρ as input. We show that for a sufficiently large horizon T, the algorithm incurs $\mathbb{R}_T(\rho) \in O(\rho^{-1} + \sqrt{(T/\rho)\log(\rho T)})$. We also provide a lower bound of $\Omega(\sqrt{T/\rho})$, establishing tightness up to a logarithmic factor with respect to T.

2. In Section 3.2, we present our second algorithm, QRM2, which does not take ρ as an input. Regardless, the algorithm achieves sub-linear ρ-reget for every $\rho > 0$. Specifically, for every $\rho > 0$ and a sufficiently large horizon T, QRM2 achieves $\mathbb{R}_T(\rho) \in o((\frac{1}{\rho}\log\frac{1}{\rho})^{2.89} + T^{0.674})$.

3. In Section 3.3, we establish a connection between (conventional) regret $\mathbb{R}^*_T$ and $\mathbb{R}_T(\rho)$. Interestingly, we find that when run on instances satisfying a common assumption made in the literature about reward distributions (Herschkorn et al., 1996, Wang et al., 2017), QRM2 also achieves sub-linear *regret* $\mathbb{R}^*_T$.

4. In Section 4, we present extensive experimental results comparing QRM2 with three separate categories of algorithms: (1) those assuming that the arms lie in a continuum (Kleinberg et al., 2008, Ray Chowdhury and Gopalan, 2017); (2) those assuming that the mean rewards come from a reservoir distribution (Wang et al., 2008); and (3) algorithms that only retain a constant number of arms in memory (Herschkorn et al., 1996, Berry et al., 1997). In the first two cases we find existing approaches to be sensitive to parameter-tuning, while our parameter-free approach shows robust performance across a variety of problem instances. Except when the arms' means indeed come from a uniform distribution (as assumed by some constant-memory algorithms), QRM2 also outperforms algorithms from the third category.

We survey the literature on regret minimisation in infinite-armed bandits before presenting our algorithms.

2 RELATED WORK

There is a vast body of literature considering regret-minimisation for infinitely many-armed bandits. Based on the assumptions they make, we classify research efforts in the area into three broad categories. We also provide brief mentions of other related work.

1. Lipschitz-continuity of mean rewards over $\mathcal{A}$. Initiated by Agrawal (1995), the "continuum-armed bandit" models a bandit whose arms come from a segment of the real line, and the rewards are a continuous function of the arms. Generalising this setting, Kleinberg (2005) and Auer et al. (2007) proposed algorithms assuming that the mean reward function $\mathbb{E}[\mathbb{M}(\cdot)] = \mu_{(\cdot)}$ is Lipschitz-continuous over the set of arms $\mathcal{A}$. Their approaches partition $\mathcal{A}$ into a finite number of intervals, treating each interval (say pulled at its middle arm (Kleinberg, 2005)) as an arm in a finite-armed bandit. The partition is progressively refined at a rate that ensures sub-linear regret. The ZOOMING algorithm proposed by Kleinberg et al. (2008), which assumes that the arms are embedded in a metric space with (known) covering dimension d, achieves a regret of $O(T^{\frac{d+1}{d+2}})$, for a horizon T. Their algorithm utilises the metric property to focus exploration on intervals potentially containing optimal arms. Understandably, the regret incurred by ZOOMING is sensitive to the definition of metric in the arms' space, and can degrade with small misspecifications of d.

A contrasting line of work follows from the work of Srinivas et al. (2010), who introduce a Gaussian Process-based algorithm, GP-UCB, for regret minimisation on infinitely many-armed bandits. Later Valko et al. (2013) proposed KERNELUCB, showing GP-UCB to be a special case. More recently, Ray Chowdhury and Gopalan (2017) have proposed two algorithms: Improved GP-UCB (IGP-UCB) and GP-Thompson sampling (GP-TS). They assume Gaussian likelihood models for the observed rewards, and Gaussian Process models for the uncertainty over reward functions. Although Ray Chowdhury and Gopalan (2017) show improved regret bounds over previous work, their algorithms are not easy to apply in practice. Foremost, the algorithms themselves give no guidance on the number of arms to explore. Also, the algorithms need several parameters to be tuned, including a confidence parameter δ, a free parameter λ, and a schedule γ_t related to information gain.

2. Particular families of reward distributions. There is a relatively large body of work that does not assume any embedding of the arms (that is, no side information), but still assumes that the distribution of mean rewards (induced by $P_{\mathcal{A}}$) comes from a particular family. Among the earliest efforts in this direction is that of Berry et al. (1997), who propose several algorithms for infinitely-many armed bandits in which the mean rewards of the arms are uniformly distributed in $[0, \mu^*]$ for some $0 < \mu^* \leq 1$. An additional assumption underlying their work is that for each arm $a \in A$, the reward distribution $M(a)$ is Bernoulli.

Wang *et al*. (2008) assume that a randomly-sampled arm's mean reward μ comes from a *reservoir distribution* $\mathcal{L}$; that is, $\exists \mu^* \in (0, 1]$ and $\nu > 0$, for which $\Pr_{\mu \sim \mathcal{L}}\{\mu > \mu^* - \epsilon\} = \Theta(\epsilon^\nu)$, for $\epsilon \to 0$. Under this assumption, they present an algorithm that incurs (1) $R^*_T = \tilde{O}(T^{1/2})$ if $\mu^* < 1$ and $\nu \leq 1$, and (2) $R^*_T = \tilde{O}(T^{\nu/(1+\nu)})$ otherwise. They have also derived lower bounds that match up to a logarithmic factor. When each arm generates Bernoulli rewards and $\mu^* = 1$, Bonald and Proutiere (2013) provide an algorithm that is optimal with the exact constant. In more recent work, Li and Xia (2017) consider a related setting in which the probability of a newly-pulled arm being near-optimal arm depends on the ratio of its expected reward to its expected *cost*, with arms having different random costs.

3. Constant-memory policies. A particular novelty of the family of algorithms studied by Berry et al. (1997) is that the algorithms maintain the reward statistics of at most one or two arms at a time. When the arms' reward distribution is *uniformly* distributed in $(0, 1)$, their algorithms are shown to achieve sub-linear regret. No closed-form upper bounds are provided when this condition does not hold. Also in the constant-memory category, Herschkorn et al. (1996) present two approaches for the problem of maximising the almost sure average reward over an infinite horizon. Although they assume that each arm generates i.i.d. Bernoulli rewards, they make no assumption on the distribution of mean rewards. They present two approaches, both of which repeatedly pull an arm until it records a certain number of successive failures. They do not provide an explicit bound on the regret.

4. Other related work. Recently, Wang et al. (2017) have proposed CEMAB, a cross-entropy based algorithm for many-armed bandit instances. Like us, they aim to focus

exploration on a small subset of arms. However, they still require the entire set of arms to be finite, which limits their experimentation to instances with a few tens of arms. They do not present any theoretical upper bounds on the regret.

The work we have discussed thus far in this section is all targeted at minimising regret. By contrast, there has also been some effort under the "pure exploration" regime to tackle infinitely-many armed bandits. For example, Carpentier and Valko (2015) aim to minimise *simple regret*, under the same assumption of a mean reservoir distribution assumption as Wang et al. (2008). Our own conception of quantile-regret is motivated by the work of Goschin et al. (2012) and Roy Chaudhuri and Kalyanakrishnan (2017), who study a PAC formulation of identifying arms above a specified reward quantile in infinitely many-armed bandits.

The primary motivation behind our work is to eliminate assumptions regarding structure and side information. Such inductive biases become counterproductive when they are not near-perfect. In Section 4, we show that the algorithms proposed by Kleinberg (2005), Ray Chowdhury and Gopalan (2017), and Wang et al. (2008), all fare poorly when their parameters are misspecified. Constant-memory algorithms, while conceptually elegant, forego the obvious benefit of retaining the statistics of multiple arms (say a few tens or hundreds) in memory. Except in tailormade settings (i.i.d. Bernoulli rewards, uniformly distributed mean rewards), our approach performs significantly better. We proceed to describe our algorithms.

3 MINIMISING QUANTILE-REGRET

At the heart of our approach for minimising quantile regret on infinitely many-armed bandit instances is to first sample out suitably-sized finite bandit instances and then to apply conventional regret minimisation algorithms on the latter. For ease of analysis, we choose the MOSS algorithm (Audibert and Bubeck, 2009) for our inner loop, since it incurs optimal (distribution-free) regret (up to a constant factor) on finite bandit instances.

First, we consider the easy case: that is, when ρ is provided as an input. Then we generalise this setting to one where ρ is not specified, and the objective is to achieve sub-linear ρ-regret for all $\rho > 0$. Our bounds will hold for sufficiently large (but still finite) horizons T. On the other hand, for a *fixed* horizon T, it is impossible to guarantee sub-linear ρ-regret for all $\rho > 0$ for all problem instances. For example, consider a problem instance with a fraction $\rho < 1/T$ of arms all being optimal, and the rest sub-optimal. T pulls will not suffice even to stumble upon an optimal arm with sufficiently high probability, let alone exploit it. The ρ-regret on such an instance will have to be linear in T.

3.1 With quantile fraction specified

In order to minimise ρ-regret for a given quantile fraction $\rho \in (0, 1]$, our primitive operation is to sample a sufficiently large number of arms using $P_{\mathcal{A}}$, and to minimise conventional regret on this set of arms by applying MOSS. We implement an "any time" algorithm by repeating this primitive procedure with progressively larger horizons, as shown in Algorithm 1.

Algorithm 1 QRM1 (with quantile fraction specified)

Require: $\mathcal{I}, \rho$
  for $r = 1, 2, 3, \cdots$ **do**
    $t_r = 2^r,\ n_r = \left\lceil \frac{1}{\rho} \max\{1, \ln \sqrt{\rho t_r}\} \right\rceil$.
    Form a set $\mathcal{K}_r$ by selecting additional $n_r - |\mathcal{K}_{r-1}|$ arms from $\mathcal{A}$ using $P_{\mathcal{A}}$, and adding them to $\mathcal{K}_{r-1}$.
    Run MOSS($\mathcal{K}_r, t_r$).
  end for

In each phase r, MOSS is called to run on a finite bandit instance with some $\mathcal{K}_r$ arms, over a finite horizon t_r. MOSS is known to incur a regret of at most $C\sqrt{|\mathcal{K}_r| t_r}$ for some constant C (Audibert and Bubeck, 2009, Theorem 5). The parameters $\mathcal{K}_r$ and t_r are specifically chosen such that with sufficiently high probability, at least one arm from $\mathcal{TOP}_\rho$ is selected in $\mathcal{K}_r$, and consequently the overall ρ-regret remains sub-linear.

Our analysis assumes $\rho \in (0, 1)$. The case of $\rho = 1$ is trivial; $\mathbb{R}_T(1)$ cannot exceed 0. For $\rho = 0$, sub-linear regret can only be achieved under the additional assumption that optimal arms have a positive probability under $P_{\mathcal{A}}$. In the analysis that follows, we use $\log$ to denote the logarithm to the base 2, and $\ln$ to denote the natural logarithm. We also take the horizon T to be a power of 2—which only changes our bounds by a constant factor.

Lemma 3.1. *For $\rho \in (0, 1)$ and for sufficiently large T,* QRM1 *achieves* $\mathbb{R}_T(\rho) = O\left(\frac{1}{\rho} + \sqrt{\frac{T}{\rho}\log(\rho T)}\right)$.

Proof. Let us consider the event during phase r that no arm from $\mathcal{K}_r$ is in $\mathcal{TOP}_\rho$. Denote this event $E_r \stackrel{\text{def}}{=} \{\mathcal{K}_r \cap \mathcal{TOP}_\rho = \emptyset\}$. We upper-bound the ρ-regret accumulated during phase r, which we denote L_r, by conditioning separately on on E_r and $\neg E_r$.

In phase r, the probability of occurrence of E_r is $\Pr\{E_r\} = (1-\rho)^{n_r}$. Now, letting $r^* = \log(e^2/\rho)$, we notice that for $r \geq r^*$, $t_r \geq e^2/\rho$, and hence we can upper bound the probability of occurrence of E_r as $\Pr\{E_r\} \leq (1-\rho)^{\rho^{-1}\ln(\sqrt{\rho t_r})} < \sqrt{1/(\rho t_r)}$. We simply take t_r as an upper bound on the phase's contribution to the regret if E_r has occurred. If E_r does not occur, then there exists at least one arm from $\mathcal{TOP}_\rho$ in $\mathcal{K}_r$. In this case, the regret is upper-bounded by $C\sqrt{n_r t_r} \leq C\sqrt{t_r \log(\rho t_r)/\rho}$, for some constant C (Audibert and Bubeck, 2009). Therefore, for

$r \geq r^*$, the ρ-regret from phase r is upper-bounded as $L_r \leq t_r \cdot \Pr\{E_r\} + C \cdot \sqrt{n_r t_r} \leq \sqrt{\frac{t_r}{\rho}} + C \cdot \sqrt{\frac{t_r}{\rho} \log(\rho t_r)} \leq C_1 \cdot \sqrt{\frac{t_r}{\rho} \log(\rho t_r)}$ for some constant C_1.

For phases $r < r^*$, the ρ-regret is trivially upper-bounded by t_r. Hence summing over all phases, we get $\mathbb{R}_T(\rho) \leq \sum_{r=1}^{r^*-1} L_r + \sum_{r=r^*}^{\log T} L_r \leq 2^{r^*} + \sum_{r=r^*}^{\log T} C_1 \cdot \sqrt{\frac{t_r}{\rho} \log(\rho t_r)}$, which is $\in O\left(\frac{1}{\rho} + \sqrt{\frac{T}{\rho} \log(\rho T)}\right)$. □

We show that this upper bound on the ρ-regret is optimal up to a logarithmic factor in the horizon. Our proof is based on a well-known lower bound for finite bandit instances (Auer et al., 2003, see Theorem 5.1).

Theorem 3.2. *[Lower bound] For every algorithm, there exists a problem instance, along with $\rho \in (0,1)$ and $T > 0$, such that $\mathbb{R}_T(\rho) \geq \min\left\{\frac{1}{20}\sqrt{\frac{T}{\rho}}, T\right\}$.*

Proof. Let ALG be any algorithm for sampling infinitely many-armed bandits. Naturally, we can also apply ALG on finite bandit instances. Given any arbitrary K-armed bandit instance, $K < \infty$, we can create a corresponding problem instance $((\mathcal{A}, M), P_{\mathcal{A}})$ wherein (1) $\mathcal{A}$ is the finite set of K arms, (2) $M(a)$ is the reward function for $a \in \mathcal{A}$, and (3) $P_{\mathcal{A}}$ samples each arm in $\mathcal{A}$ with an equal probability of $1/K$. Now, if we set $\rho = 1/K$, observe that $\mathcal{TOP}_\rho$ can only contain *optimal* arms from $\mathcal{A}$, and hence, the ρ-regret incurred by ALG on $((\mathcal{A}, M), P_{\mathcal{A}})$ is the same as the conventional regret on the original finite instance.

Suppose, contrary to the statement of the theorem, ALG is such that for all input problem instances, for all $\rho \in (0,1)$ and for all $T > 0$, its ρ-regret satisfies $\mathbb{R}_T(\rho) < \min\left\{\frac{1}{20}\sqrt{\frac{T}{\rho}}, T\right\}$. From the translation described above, it follows that ALG incurs $\mathbb{R}^*_T < \min\{\frac{1}{20}\sqrt{KT}, T\}$ for all finite K-armed bandit instances, $K > 0$ and horizons $T > 0$. However, Auer et al. (2003, see Theorem 5.1) have shown that no such algorithm exists for finite instances. Our proof is complete. □

3.2 With quantile fraction not specified

We can now drop the requirement that ρ is given to the algorithm as an input. Rather, we iteratively optimise ρ-regret for progressively decreasing values of ρ.

Algorithm 2 follows the same template as Algorithm 1, except that the number of arms to sample in each phase r is set to be a polynomial function of t_r, with the power $\alpha = 0.347$ set to minimise the ρ-regret's dependence on T.

Although QRM2, the algorithm specified above, does not require any knowledge of ρ, we shall analyse its ρ-regret for some fixed $\rho > 0$.

Algorithm 2 QRM2 (with quantile fraction not specified)

Require: $\mathcal{I}$
  Set $\alpha = 0.347$ and $\mathcal{K}_0 = \emptyset$.
  for $r = 1, 2, 3, \cdots$ **do**
    $t_r = 2^r,\ n_r = \lceil t_r^\alpha \rceil$.
    Form a set $\mathcal{K}_r$ by selecting additional $n_r - |\mathcal{K}_{r-1}|$ arms from $\mathcal{A}$ using $P_{\mathcal{A}}$, and adding to $\mathcal{K}_{r-1}$.
    Run MOSS($\mathcal{K}_r, t_r$).
  end for

Theorem 3.3. *[Sub-linear quantile-regret of* QRM2*] For $\rho \in (0,1)$ and for sufficiently large T,* QRM2 *incurs $\mathbb{R}_T(\rho) \in o\left(\left(\frac{1}{\rho}\log\frac{1}{\rho}\right)^{2.89} + T^{0.674}\right)$.*

Proof. Considering some fixed $\rho \in (0,1)$, we upper-bound the ρ-regret in two parts: (1) when no arms from $\mathcal{TOP}_\rho$ are chosen, and (2) when at least one arm is chosen. To analyse the first part, we show that for $r^* \stackrel{\text{def}}{=} \lceil (1/\alpha)\log((1/\rho)\log(1/\rho)) \rceil$, if $r \geq r^*$, then $\mathcal{K}_r$ is sufficiently large to contain an arm from $\mathcal{TOP}_\rho$ with high probability. To show that, like before, we define the event that no arm from $\mathcal{TOP}_\rho$ is in $\mathcal{K}_r$ as $E_r(\rho) \stackrel{\text{def}}{=} \{\mathcal{K}_r \cap \mathcal{TOP}_\rho = \emptyset\}$. It follows $\Pr\{E_r(\rho)\} = (1-\rho)^{n_r}$. Now, for $r \geq r^*$, using Lemma 5.1 (provided in Appendix A[1]), we get $\Pr\{E_r(\rho)\} \leq \exp(-\lceil(\alpha(1+\gamma)^{-1} \cdot \ln t_r^{\log \mathrm{e}}\rceil) \leq t_r^{-\alpha \log \mathrm{e}/(1+\gamma)}$. Hence, if the algorithm runs for T pulls, then the regret due to occurrence of $E_r(\rho)$ is upper bounded as

$$\sum_{r=1}^{\log T} t_r \Pr\{E_r(\rho)\} \in O\left(t_{r^*} + T^{1-\frac{\alpha \log \mathrm{e}}{1+\gamma}}\right). \quad (3)$$

Now we analyse the second part: that is, upper-bounding the regret incurred if at least one from the $\mathcal{TOP}_\rho$ is in $\mathcal{K}_r$ (the event $\neg E_r(\rho)$). Let us assume that C is a constant such that the regret incurred by MOSS in phase r (given $\neg E_r(\rho)$) is at most $C\sqrt{n_r t_r} = C t_r^{(1+\alpha)/2}$. Therefore, assuming total number of pulls as T, the total regret incurred on r^*-th phase onward is upper bounded as

$$\sum_{r=r^*}^{\log T} C\sqrt{n_r t_r} \leq C' T^{(1+\alpha)/2} \quad (4)$$

for some constant C'. The intermediate steps to obtain (3) and (4) are shown in Appendix-A. Combining (3) and (4), and substituting for t_{r^*}, we get $\mathbb{R}_T(\rho) = O\left(\left(\frac{1}{\rho}\log\frac{1}{\rho}\right)^{\frac{1}{\alpha}} + T^{1-\frac{\alpha\log\mathrm{e}}{1+\gamma}} + T^{(1+\alpha)/2}\right)$. We conclude by noticing that $\alpha = 0.5/(0.5 + \log\mathrm{e}/(1+\gamma)) \approx 0.3466$ minimises $\mathbb{R}_T(\rho)$ with respect to T. □

[1] Appearing at the end of the extended version of this paper at https://www.cse.iitb.ac.in/~shivaram/papers/rk_uai_2018.pdf.

The upper bound in Theorem 3.3 cannot be directly compared with regret bounds in the literature (Kleinberg, 2005, Wang et al., 2008) since our bound is on the ρ-regret. As yet, we do not know if the dependence of ρ-regret on the horizon T can be improved. Even so, the sub-linear upper bound we have shown on $\mathbb{R}_T(\rho)$ assumes a special significance in the study of infinitely-many armed bandits. Observe that the upper bound holds for every $\rho > 0$ and for *every* bandit instance. By contrast, conventional regret-minimisation (of $\mathbb{R}^*_T$) cannot assure sub-linear regret unless the bandit instance itself satisfies additional assumptions (of which several have been made in the literature). By taking $\rho > 0$, we have chosen to change our objective (albeit slightly), rather than place demands on the input bandit instance.

Interestingly, we find that on bandit instances that do satisfy a standard assumption made to achieve sub-linear $\mathbb{R}^*_T$, QRM2, too, achieves sub-linear $\mathbb{R}^*_T$, in spite of being designed to minimise $\mathbb{R}_T(\rho)$ for $\rho > 0$. We proceed to discuss this connection between $\mathbb{R}_T(\rho)$ and $\mathbb{R}^*_T$.

3.3 Quantile-regret and conventional regret

Given a problem instance $((\mathcal{A}, M), P_{\mathcal{A}})$, we first show that minimising $\mathbb{R}^*_T$ is sufficient to minimise $\mathbb{R}_T(\rho)$ for all $\rho > 0$, but the converse is not true.

Lemma 3.4. *For any algorithm and input problem instance, if $\mathbb{R}^*_T \in o(T)$, then it must hold that $\mathbb{R}_T(\rho) \in o(T)$, for all $\rho > 0$. However, the converse is not true.*

Proof. For the first part, (1) is written as $\mathbb{R}^*_T = T \cdot (\mu^* - \mu_\rho) + T \cdot \mu_\rho - \sum_{t=1}^{T} \mathbb{E}[\mu_t] = T \cdot (\mu^* - \mu_\rho) + \mathbb{R}_T(\rho)$. Hence, $\mathbb{R}^*_T \in o(T) \implies \mathbb{R}_T(\rho) \in o(T)$, for all $\rho \in [0, 1]$.

The second part is obtained by considering an infinitely-many armed bandit instance with finitely many optimal arms. Formally, take $|\mathcal{A}| = \infty$ and $P_{\mathcal{A}}$ to be the uniform distribution over $\mathcal{A}$. Let $S \subset \mathcal{A}$, such that $\forall a \in S$, $\mu_a = \mu^*$, $|S| < \infty$, and $\forall a \in \mathcal{A} \setminus S$: $\mu_a = \bar{\mu} < \mu^*$. Now, S being finite, with probability 1, no arm from S will be picked by $P_{\mathcal{A}}$. Therefore, for $\rho > 0$, $\mu_\rho = \bar{\mu}$, and so $\mathbb{R}^*_T = T \cdot (\mu^* - \bar{\mu}) + \mathbb{R}_T(\rho) \geq T \cdot (\mu^* - \bar{\mu}) \in \Omega(T)$. □

Although in general, achieving sub-linear ρ-regret does not imply achieving sub-linear regret, this turns out asymptotically true for QRM2 on the family of bandit instances considered by Wang et al. (2008), of which the family of instances considered by Berry et al. (1997) is a subset. In these instances, the distribution of the mean reward μ, denoted $D(\mu)$, has the "reservoir" property, as detailed below.

Proposition 3.5 (Case Study)**.** QRM2 *achieves $\mathbb{R}^*_T \in o(T)$ as $T \to \infty$, under the assumption that $\Pr_{\mu \sim D(\mu)}\{\mu > \mu^* - \epsilon\} = \Theta(\epsilon^\nu)$, for $\epsilon \to 0$, where ν is a positive constant.*

Proof. The assumption amounts to the existence $\rho_0 \in (0, 1]$ such that for $0 < \rho < \rho_0$, $c_l(\mu^* - \mu_\rho)^\nu \leq \rho \leq c_u(\mu^* - \mu_\rho)^\nu$, where c_u, c_l are positive constants. Defining $h(\rho) \stackrel{\text{def}}{=} \mu^* - \mu_\rho$, we see that for $\rho \in (0, \rho_0]$: $h(\rho) \leq (\rho/c_l)^{1/\nu}$.

We know from Theorem 3.3 that for any given $\rho > 0$, and for a sufficiently large horizon T, QRM2 achieves $\mathbb{R}_T(\rho) \in o(T)$. Equivalently, for every sufficiently large T, there exists a $\rho(T) \in (0, 1]$ such that, for all $\rho \geq \rho(T)$, QRM2 achieves $\mathbb{R}_T(\rho) \in o(T)$. We also notice that $\rho(T)$ is a monotonic non-increasing sequence that converges to 0. Hence, there exists a sufficiently large horizon T_0 such that for all $T \geq T_0$, $\rho(T) \leq \rho_0$. In other words, for horizon $T > T_0$, $h(\rho(T)) \leq (\rho(T)/c_l)^{1/\nu}$. Since $\lim_{T\to\infty}(\rho(T)/c_l)^{1/\nu} = 0$ and $h(\rho(T)) \geq 0$, we get $\lim_{T\to\infty} h(\rho(T)) = 0$. Since $\mathbb{R}^*_T = T \cdot h(\rho(T)) + \mathbb{R}_T(\rho)$, we get $\lim_{T\to\infty} \frac{\mathbb{R}^*_T}{T} = \lim_{T\to\infty} \left(h(\rho(T)) + \frac{\mathbb{R}_T(\rho(T))}{T} \right) = 0$, which means that $\mathbb{R}^*_T$ is asymptotically $o(T)$. □

4 EXPERIMENTS AND RESULTS

In this section, we compare QRM2 with competing approaches for regret minimisation on infinitely many-armed bandits. We consider representative algorithms from the categories described in Section 2, and investigate the efect of their parameters, in tandem with a comparison with QRM2. Although QRM2 is designed to minimise ρ-regret for progressively decreasing ρ values, we use conventional regret as the evaluation metric in all our experiments. This choice essentially amounts to evaluating the total reward accrued over a given horizon, which is perhaps the most relevant measure in practice. In all our experiments, the reward distributions of arms are Bernoulli. Note that both ZOOMING (Kleinberg et al., 2008) and QRM2 proceed through phases, progressively doubling the phase length. To improve sample efficiency, we retain the statistics of pulls from previous phases and correspondingly adjust the "budget" term in the confidence bound.

4.1 Comparison with ZOOMING

The ZOOMING algorithm (Kleinberg et al., 2008) works on a bandit instance comprising a set of arms $\mathcal{A} = [0, 1]$, with the expected mean reward $\mu_{(\cdot)}$ being Lipschitz-continuous over $\mathcal{A}$. The metric defined on $\mathcal{A}$ is: for $x, y \in \mathcal{A}$, $L_d(x, y) = |x - y|^{1/d}$, where $d \geq 1$ is a *known*, user-specified parameter. For a given horizon T, ZOOMING is shown to incur a regret of $\tilde{O}(T^{(d+1)/(d+2)})$. The algorithm proceeds by maintaining confidence bounds on the mean rewards of contiguous regions of $\mathcal{A}$; a new region is created whenever the existing ones fail to cover some portion of $\mathcal{A}$. In our implementation, a new region is created by picking an uncovered region uniformly at random. Its "centre" is picked uniformly at random from the points it contains.

We compare QRM2 with ZOOMING on four problem instances, shown in Figure 2 and specified in Appendix-B. On each instance we compare the cumulative regret of QRM2

with that of ZOOMING for $d \in \{1, 2\}$. The results at different horizons are presented in Figure 3. Foremost, observe that the performance of ZOOMING is fairly sensitive to d: on instances I-P and I-S, the variant with $d = 1$ performs noticeably better; on instances I-N and I-W, the variant with $d = 2$ is superior. On the instance I-N the better of these two variants performs close to QRM2. On an unknown problem instance, it is unrealistic to expect that the user will be able to guess a good value for d beforehand.

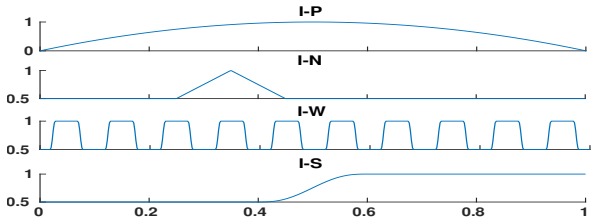

Figure 2: Four problem instances (fully specified in Appendix-B). In all four cases, $\mathcal{A} = [0, 1]$, as shown on the x axis. The y axis shows the mean reward μ_a for $a \in \mathcal{A}$. QRM2 takes $P_{\mathcal{A}}$ to be the uniform distribution over $\mathcal{A}$.

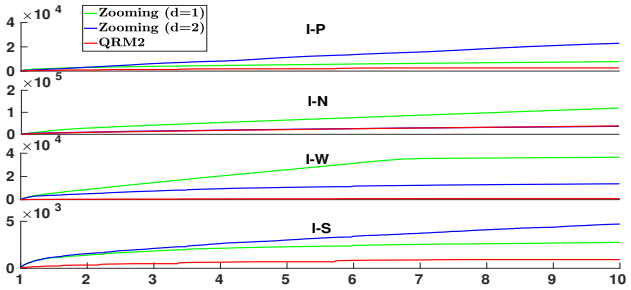

Figure 3: Cumulative regret (y axis) incurred by ZOOMING and QRM2 on the instances in Figure 2. The x axis shows the horizon / 10^5. Each plot is an average of 100 runs.

A second limitation in practice arises from the quality of the features used to generalise across $\mathcal{A}$, which effectively determine the Lipschitz constant of the mean reward function. Instances I-W and I-S have exactly the same mean distribution D—they only differ in the indexing of the arms, which, in practice, would depend on the feature representation. The regret of ZOOMING on these instances (whether with $d = 1$ or with $d = 2$) varies at least three-fold. By ignoring the arm indices and the metric, QRM2 registers exactly the same regret on both instances.

4.2 Comparison with Gaussian Process algorithms

In our next set of experiments, we compare QRM2 with IGP-UCB and GP-TS (Ray Chowdhury and Gopalan, 2017). Our experiments are run on the light sensor data set[2], on which the authors have themselves benchmarked their methods. We refer the reader to the original paper for a description of the data set (Ray Chowdhury and Gopalan, 2017, see Section 6), which encodes bandit instances with arms corresponding to sensors. We run IGP-UCB and GP-TS with the same parameters used by the authors.

From Table 1, we find that GP-TS outperforms IGP-UCB, exactly as reported by Ray Chowdhury and Gopalan (2017). However, QRM2 outperforms both GP-TS and IGP-UCB by a large margin. We posit as one reason for the efficiency of QRM2 its use of MOSS as the underlying regret minimisation procedure. On the other hand, using Gaussian Processes to generalise over the space of arms would only work well if nearby arms indeed have similar mean rewards. Without good generalisation, the confidence bounds resulting from Gaussian Processes are likely to be loose, and therefore a poor guide for exploration.

Table 1: Cumulative regret after 10^6 pulls, averaged over 192 test instances, with one standard error.

Algorithm	Average cumulative regret
GP-TS	$2.58 \times 10^4 \pm 36.75 \times 10^2$
IGP-UCB	$3.86 \times 10^5 \pm 18.05 \times 10^4$
QRM2	$\mathbf{0.14} \times 10^4 \pm 0.13 \times 10^2$

4.3 Comparison with algorithm of Wang et al. (2008)

We compare QRM2 with the algorithm of Wang et al. (2008) for unspecified horizons. Recall that the sub-linear regret bounds shown by Wang et al. (2008) are based on the assumption that the mean distribution is a "reservoir": that is, $\Pr_{\mu_a \sim P_{\mathcal{A}}}\{\mu_a > \mu^* - \epsilon\} = \Theta(\epsilon^\nu)$, for $\epsilon \to 0$. We notice that for $\mu^* \in (0, 1]$ and $\nu > 0$, $f(\mu) = \frac{\nu}{\mu^{*\nu}}(\mu^* - \mu)^{\nu-1}$ is a density function of some reservoir distribution. This follows from the fact that CDF of $f(\mu)$ is given by $F(\mu) = 1 - \frac{1}{\mu^{*\nu}}(\mu^* - \mu)^\nu$. Therefore $\Pr_{\mu \sim f(\mu)}\{\mu > \mu^* - \epsilon\} = 1 - F(\mu) \in \Theta(\epsilon^\nu)$. It is worth noting that for $\nu = 1$, $f(\mu)$ is the uniform distribution.

The any-time algorithm given by Wang et al. (2008) requires three parameters: (1) an exploration rate ξ_t, for the t-th pull, such that $2\ln(10 \ln t) \le \xi_t \le \ln t$, (2) the shape parameter ν, and (3) whether $\mu^* = 1$. We refer the reader to the original paper (Wang et al., 2008, see Section 2) for a full specification. We test this algorithm along with QRM2 on four problem instances, shown in Table 2. In all cases, $\mathcal{A} = [0, 1]$, and we take $P_{\mathcal{A}}$ as the uniform distribution over $\mathcal{A}$. Each problem instance is such that its optimal mean μ^* is either 1 or 0.6, and its reward distribution $D(\mu)$ is either $\beta(0.5, 2)$ or $\beta(1, 1)$ (scaled to have support in $[0, \mu^*]$). The algorithm of Wang et al. (2008)'s needs to be supplied ν such that the complementary cumulative distribution func-

[2] www.cs.cmu.edu/~guestrin/Class/10708-F08/projects/lightsensor.zip

tion (CCDF) of $f(\mu)$ will overestimate that of $D(\mu)$ beyond some $\mu_0 < \mu^*$. Also, as the algorithm needs to know whether of not $\mu^* = 1$, we supply a representative value $\mu^\#$ for μ^*. Table 2 explicitly shows the parameterisation used for the different instances, and Figure 4 depicts the CCDF of the corresponding $D(\mu)$, and that of $f(\mu)$ for different values of ν. In practice, an experimenter might not have a precise estimate of (ν, μ^*), and hence the values supplied might not meet the above criteria. In Table 2, depending on whether or not the values of $\nu, \mu^\#$ respect the criteria, the corresponding cells are marked by ✓and ✗, respectively. Also, the values of $\nu, \mu^\#$ (from our set) for which the CCDF of $f(\mu)$ coincides with or fits $D(\mu)$ most closely are marked Exact and Closest, respectively.

Table 2: Summary of problem instances used in Section 4.3, along with different parameterisations of the algorithm of Wang et al. (2008). For explanations see Section 4.3.

Instances	μ^*	$D(\mu) = \beta(a, b)$		$\mu^\#$		ν		
		a	b	1.0	0.6	0.4	1	2
I-1	1	0.5	2	✓	✗	✓	✓	✓Closest
I-2	1	1	1	✓	✗	✓	✓Exact	✗
I-3	0.6	0.5	2	✗	✓	✓	✓	✓Closest
I-4	0.6	1	1	✗	✓	✓	✓Exact	✗

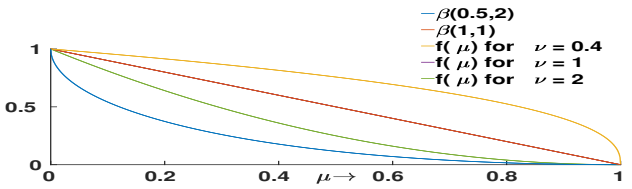

Figure 4: Complementary CDF (CCDF) values (y axis) for various distributions. The CCDF for $\beta(1, 1)$ coincides with that of $f(\mu)$ for $\nu = 1$.

For a horizon of 10^6, QRM2, which is parameter-free, explores a fixed number of arms ($= 94$). On the other hand, Wang *et al.*'s algorithm explores 10^4 arms for $\nu = 2$. For $\nu = 0.4$ it explores 16 and 52 arms for $\mu^\# = 0.6$ and $\mu^\# = 1$, respectively. Figure 5 shows that in spite of providing values of $(\nu, \mu^\#)$ that closely (or even exactly) track $D(\mu)$, QRM2 outperforms Wang *et al.*'s algorithm by a significant margin. We note that optimistic values of these parameters helps their result improve, but incorrect parameterisation severely degrades performance. Another non-trivial factor in the performance of their algorithm is the exploration rate ξ_t. While varying ξ_t within their prescribed range keeps the regret upper bound unaffected, in practice it is observed to have a significant effect on regret. In Algorithm 2, we set the α parameter of QRM2 to 0.347 to optimise a *theoretical* bound. In practice, tuning α for different problem instances further improves QRM2's performance. In line with our intent to not depend on tuning, we refrain from reporting these optimised results.

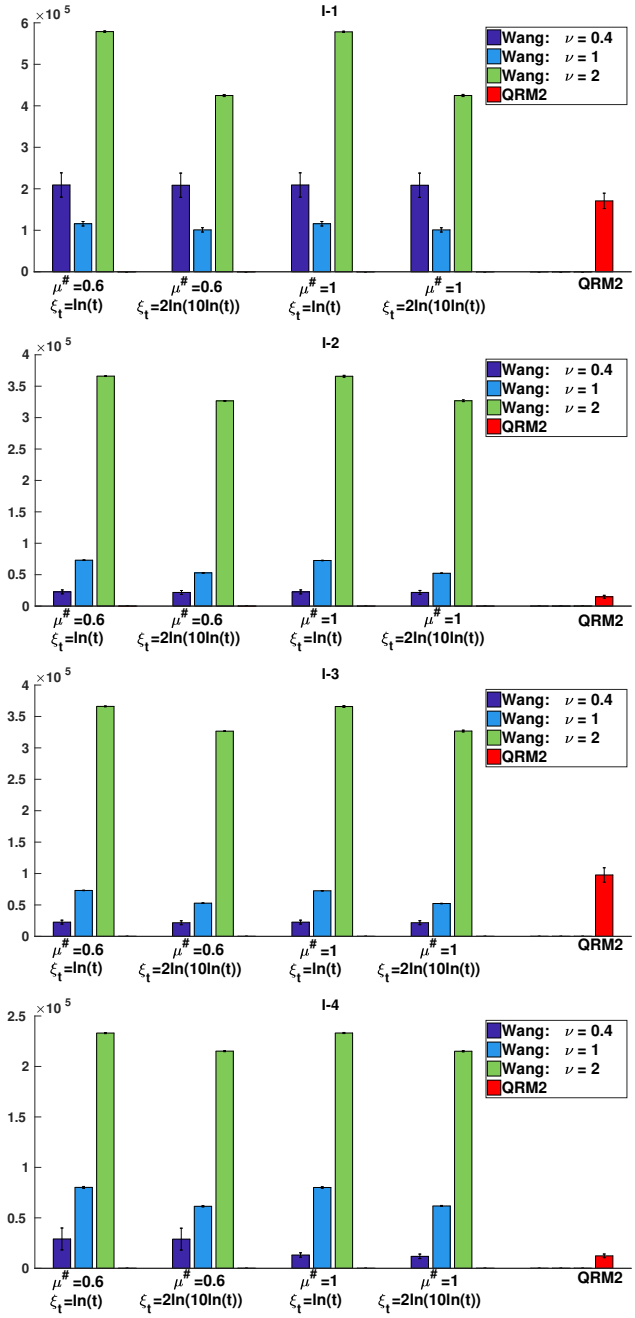

Figure 5: Cumulative regret incurred by QRM2 and the algorithm of Wang et al. (2008) after 10^6 pulls on the instances in Table 2. Each bar is an average of 20 runs, and shows one standard error bar. The accompanying parameters are explained in Section 4.3.

4.4 Comparison with constant-memory algorithms

Recall that the algorithms of Herschkorn et al. (1996) and Berry et al. (1997) keep the statistics of only a single arm (or two) in memory. They are specifically designed for bandit instances that yield Bernoulli rewards. The "Non-stationary" algorithm of Herschkorn et al. (1996) repeatedly pulls the i-th arm, for $i = 1, 2, \ldots$, until it produces i consecutive failures—at which point a new arm is pulled. Berry et al. (1997) propose three strategies for problem instances in which the distribution of means is uniform over $[0, \mu^*]$ for some $\mu^* \in [0, 1]$. These strategies assume that the horizon T is given. For example, the "$\sqrt{T}$-run" switches out the current arm upon a single failure, unless the arm produces $\sqrt{T}$ successes (in which case it is pulled for the remaining horizon). If $\sqrt{T}$ arms have been pulled and discarded, the arm with the highest observed empirical mean thus far is pulled for the remainder of the run. The "$\sqrt{T}\ln T$-learning" strategy and the "Nonrecalling $\sqrt{T}$-run" are variants built around a similar theme; we refer the reader to the original paper (Berry et al., 1997) for precise specifications. Table 3 presents a comparison of incurred cumulative regret on the instances I-1, I-2, I-3 and I-4. On I-1, QRM2 outperforms all the other strategies by a significant margin. This result is not surprising, since (1) QRM2 uses additional memory (94 arms for a horizon of 10^6), and (2) unlike the strategies of Berry et al. (1997), it does not assume that the mean rewards are uniformly distributed. On I-2, which indeed has uniformly-distributed means, the Nonrecalling $\sqrt{T}$-run strategy of Berry *et al*. performs marginally better than QRM2. However, this win comes at the expense of incurring very high regret on I-1, in which near-optimal arms are less likely to be encountered. Interestingly, on I-3 and I-4, the Non-stationary policy of Herschkorn et al. (1996) policy outperforms all three from Berry et al. (1997). Yet, all these algorithms are outperformed by QRM2.

Table 3: Cumulative regret ($/10^5$) of QRM2 and strategies proposed by Herschkorn et al. (1996) and Berry et al. (1997) after 10^6 pulls, on instances I-1, I-2, I-3 and I-4. Each result is the average of 20 runs, showing one standard error.

Algorithms	I-1	I-2	I-3	I-4
Non-stationary Policy (Herschkorn et al., 1996)	3.58 ±0.4	1.11 ±0.2	1.64 ± 0.2	0.79 ± 0.1
$\sqrt{T}$-run (Berry et al., 1997)	6.18 ±0.5	1.11 ±0.4	4.18 ± 0.3	2.03 ± 0.3
$\sqrt{T}\ln T$-learning (Berry et al., 1997)	6.32 ±0.4	0.69 ±0.3	4.38 ± 0.2	2.15 ± 0.3
Nonrecalling $\sqrt{T}$-run (Berry et al., 1997)	5.35 ±0.5	**0.03** ±0.004	4.56 ± 0.001	2.55 ± 0.001
QRM2	**1.71** ±0.2	0.15 ±0.02	**0.98** ± 0.1	**0.12** ± 0.01

5 CONCLUSION

In this paper, we present an approach to manage the explore-exploit trade-off in bandit instances that contain many more arms than the possible number of experiments. While most existing approaches in this setting assume special properties of the arms' reward function or some structure over the set of arms, we make no such assumptions. Rather, we reformulate the problem by introducing the notion of quantile regret (or ρ-regret), which is defined with respect to the $(1-\rho)$-th quantile of the mean reward distribution—unlike conventional regret, which is defined with respect to the highest mean. We present sub-linear upper bounds on the ρ-regret when (1) ρ is specified to the algorithm, and (2) ρ is not specified to the algorithm. We also prove that our QRM2 algorithm, although it is designed to minimise ρ-regret for small ρ, indeed achieves sub-linear regret under the assumption that the instance's mean rewards come from a reservoir distribution (Wang et al., 2008).

We provide extensive empirical justification for quantile-regret minimisation. Our experiments show that the ZOOMING algorithm (Kleinberg et al., 2008) is sensitive to the given metric and the Lipschitz-continuity of the reward function. With slight perturbations to its parameters, the algorithm incurs a significantly higher regret. The GP-TS, and IGP-UCB algorithms (Ray Chowdhury and Gopalan, 2017) do not explicitly specify the number of arms to explore. Both algorithms perform much worse than QRM2 on the light sensor problem. We find that even when specified the exact distributional parameter, the algorithm proposed by Wang et al. (2008) can incur a higher regret than QRM2. It is infeasible in practice to know the optimal parameter setting for a given problem instance, and it is undesirable to have to find the right parameters using techniques such as cross-validation. The parameter-free approach taken by QRM2 makes it especially appealing to implement as a baseline across different domains and problem instances.

The constant-memory policies proposed by Herschkorn et al. (1996) and Berry et al. (1997) are akin to QRM2 in being simple and parameter-free. On the theoretical side, it seems plausible that their dependence on uniformly-distributed rewards can be removed in lieu of providing a finite time upper bound on the quantile-regret (rather than asymptotic guarantees). Practically, it also seems appealing to generalise these methods to work with larger, even if constant, memory sizes. In future work, we plan to analyse constant-memory policies within the framework of quantile-regret minimisation. We also aim to examine possible improvements to both the upper and lower bounds presented in this paper.

Acknowledgements

We thank Sayak Ray Chowdhury for sharing code and providing useful guidance. SK was partially supported by SERB grant ECR/2017/002479.

References

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control Optim.*, 33(6):1926–1951, 1995.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proc. COLT 2009*, pages 217–226, 2009. URL `https://hal-enpc.archives-ouvertes.fr/hal-00834882/file/COLT09a.pdf`.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2003.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.

Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *Proc. COLT 2007*, pages 454–468. Springer, 2007.

D.A. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Chapman & Hall, 1985.

Donald A. Berry, Robert W. Chen, Alan Zame, David C. Heath, and Larry A. Shepp. Bandit problems with infinitely many arms. *The Annals of Stat.*, 25(5):2103–2116, 1997.

Thomas Bonald and Alexandre Proutiere. Two-target algorithms for infinite-armed bandits with Bernoulli rewards. In *Adv. NIPS 26*, pages 2184–2192. Curran Associates, Inc., 2013.

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *Proc. ICML 2015*, pages 1133–1141. JMLR, 2015.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proc. AISTATS 2011*, volume 15, pages 208–214. PMLR, 2011.

David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann publishers Inc., 3rd edition, 2002.

Avishek Ghosh, Sayak Ray Chowdhury, and Aditya Gopalan. Misspecified linear bandits. In *Proc. AAAI 2017*, pages 3761–3767. AAAI Press, 2017.

Sergiu Goschin, Ari Weinstein, Michael L. Littman, and Erick Chastain. Planning in reward-rich domains via PAC bandits. In *Proc. EWRL 2012*, volume 24, pages 25–42. JMLR, 2012.

Stephen J. Herschkorn, Erol Pekz, and Sheldon M. Ross. Policies without memory for the infinite-armed Bernoulli bandit under the average-reward criterion. *Prob. in the Engg. and Info. Sc.*, 10(1):21–28, 1996.

Yoshiaki Kadono and Naoki Fukuta. Lakube: An improved multi-armed bandit algorithm for strongly budget-constrained conditions on collecting large-scale sensor network data. In *PRICAI 2014: Trends in Artificial Intelligence*, pages 1089–1095. Springer International Publishing, 2014.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Adv. NIPS 17*, pages 697–704. MIT Press, 2005.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proc. STOC 2008*, pages 681–690. ACM, 2008.

Haifang Li and Yingce Xia. Infinitely many-armed bandits with budget constraints. In *Proc. AAAI 2017*, pages 2182–2188. AAAI Press, 2017.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proc. ICML 2017*, volume 70, pages 844–853. PMLR, 2017.

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. PAC identification of a bandit arm relative to a reward quantile. In *Proc. AAAI 2017*, pages 1977–1985. AAAI Press, 2017.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML 2010*, pages 1015–1022. Omnipress, 2010.

Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proc. CIKM 2013*, pages 1587–1594. ACM, 2013.

Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artif. Intl.*, 214:89–111, 2014.

Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Proc. UAI 2013*, pages 654–663. AUAI Press, 2013.

Erli Wang, Hanna Kurniawati, and Dirk P. Kroese. CEMAB: A cross-entropy-based method for large-scale multi-armed bandits. In *Artif. Life and Computnl. Intl.*, pages 353–365. Springer Intl. Publishing, 2017.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Adv. NIPS 21*, pages 1729–1736. Curran Associates Inc., 2008.