Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Tight Policy Regret Bounds for Improving and Decaying Bandits

Hoda Heidari
University of Pennsylvania
hoda@seas.upenn.edu

Michael Kearns
University of Pennsylvania
mkearns@cis.upenn.edu

Aaron Roth
University of Pennsylvania
aaroth@cis.upenn.edu

Abstract

We consider a variant of the well-studied multi-armed bandit problem in which the reward from each action evolves monotonically in the number of times the decision maker chooses to take that action. We are motivated by settings in which we must give a series of homogeneous tasks to a finite set of arms (workers) whose performance may improve (due to learning) or decay (due to loss of interest) with repeated trials. We assume that the arm-dependent rates at which the rewards change are unknown to the decision maker, and propose algorithms with provably optimal *policy regret* bounds, a much stronger notion than the often-studied external regret. For the case where the rewards are increasing and concave, we give an algorithm whose policy regret is sublinear and has a (provably necessary) dependence on the time required to distinguish the optimal arm from the rest. We illustrate the behavior and performance of this algorithm via simulations. For the decreasing case, we present a simple greedy approach and show that the policy regret of this algorithm is constant and upper bounded by the number of arms.

1 Introduction

In the well-studied multi-armed bandit setting, a decision maker is faced with the problem of choosing which arm to play over a sequence of trials. Each time the decision maker pulls an arm, he receives a reward, and his goal is to minimize some notion of *regret*. The majority of previous studies consider the so-called *external regret* as the objective. External regret is the difference between the total reward collected by an online algorithm and the maximum reward that could have been collected by pulling a single arm *on the same sequence of rewards as the one generated by the algorithm*. Past solutions to this problem can be divided into two main categories: first, solutions that rely on statistical assumptions about the underlying reward process (see for instance[Robbins, 1985; Gittins *et al.*, 2011; Auer *et al.*, 2002]); and second, solutions for the setting where an adversary, who is capable of reacting to the choices of the decision maker, determines the sequence of rewards (see e.g. [Auer *et al.*, 2002]).

Our goal in this work is to minimize a stronger notion of regret, namely *policy regret*, in a setting where the reward from each arm changes monotonically[1] every time the decision maker pulls that arm. More precisely, in the model we propose, there exist n arms, each with a *reward curve* that is unknown to the decision maker. Every time the decision maker pulls an arm, the corresponding reward of that arm monotonically changes according to its underlying reward curve.

Unlike traditional statistical approaches, we do not make probabilistic assumptions on the behavior of the arms; and unlike traditional adversarial approaches, we do not assume a fixed sequence of payoffs against which we measure our regret. In particular, under our assumptions the algorithm itself is actually *generating* the sequence of payoffs via its decisions. The right notion of regret is thus not comparing to the best single arm in hindsight on the sequence generated (external regret), but to the *best sequence that could have been generated* — that is, to the optimal policy that knows the reward functions. This stronger notion is what is called *policy regret* in the literature [Arora *et al.*, 2012]. The following simple example further illustrates the difference between policy and external regret.

Example 1 *Consider a setting in which there are two arms and time horizon $T \gg 10$. Arm 1 returns a reward of $\frac{i}{T}$ when pulled for the ith time, and arm 2 always returns a reward of 0.1. Consider the algorithm that always pulls arm 2. The external regret of this algorithm is zero because at every time step, it pulls the arm that would give it the largest reward on that time step. But the policy regret of this algorithm grows linearly with T, as the best policy in hindsight indeed pulls arm 1 at every time step.* [2]

We are motivated by settings in which the available actions correspond to the assignment of a sequence of relatively homogeneous tasks to one of a finite set of workers, as is often the case in crowdsourcing systems (see for example [Tran-Thanh *et al.*, 2014; Heidari and Kearns, 2013]). In such settings, it is reasonable to expect that workers' performance

[1] See [Slivkins, 2011] for other motivating examples and a different formalization of monotone bandits.

[2] It is easy to adapt this example to show that specific algorithms with no external regret, such as EXP3, will indeed fail to have small policy regret.

may improve or decay with repeated trials, depending on the nature of the task. For example, tasks that are unfamiliar and challenging (such as segmenting brain images into individual neurons [Seung, 2015]) may require a training period during which performance gradually improves. In contrast, in a task primarily requiring only human perception (such as the transcription of license plates in images [Barowy *et al.*, 2012; Du *et al.*, 2013]), subjects may immediately be able to perform the task at a high level, but its tedious and taxing nature may lead to performance decay with increased workload. In both cases, different subjects may have different rates of improvement or decay.

The rest of this paper is organized as follows: In Section 2, we introduce the model. In Section 3, we study the case of increasing and concave reward functions and present an optimal online algorithm whose policy regret depends on a parameter we call τ. This parameter quantifies the time required to distinguish the optimal arm from the others, and we prove the dependence is necessary. We prove that the policy regret of this algorithm is sublinear and further illustrate the range of behaviors of τ and the performance of the algorithm via simulations. Finally, in Section 4 we investigate the case of decreasing rewards and present a provably optimal algorithm whose policy regret is constant and upper bounded by the number of arms.

1.1 Related Work

As discussed in [Arora *et al.*, 2012], the notion of external regret fails to capture the actual regret of an online algorithm compared to the optimal sequence of actions it *could* have taken when the adversary is adaptive. Policy regret is defined to address this counterfactual setting. The authors show in [Arora *et al.*, 2012] that if the adaptive adversary has *bounded memory*, then a variant of traditional online learning algorithms can still guarantee no policy regret. In our work, the rewards depend on *all* the actions taken so far and as the result our model is not captured by the bounded memory setting.

While our model is not captured by any of the previous papers in the bandit literature, the following papers are conceptually related. [Tekin and Liu, 2012] study a setting in which the reward from each arm is modeled as a finite-state Markov chain. Authors consider two cases: *rested* and *restless* arms. In the rested case, the state of each underlying Markov chain remains frozen unless the corresponding arm is played. While in our setting the arms are indeed rested, the reward process we study cannot be modeled by a *finite*-state Markov chain (see also [Neu *et al.*, 2014]). [Gabillon *et al.*, 2013] studies the problem of picking a subset of arms at each step in a setting where the reward is a submodular function of the chosen subset. [Streeter *et al.*, 2009] studies the problem of assigning items to K positions such that a submodular utility function is maximized. Finally, our model is somewhat related to the line of research on best arm identification, however, previous studies on this topic mainly rely on stochastic assumptions on the rewards (see for example [Audibert and Bubeck, 2010],[Chandrasekaran and Karp, 2012]), and they do not apply to our setting.

There is much work in the psychology literature studying the human learning process on cognitive tasks (see e.g. [Atkinson *et al.*, 1965; Mangal, 2009]), but to our knowledge ours is the first to model it in a bandits setting. There are however, a limited number of papers addressing relevant questions from a heuristic or empirical viewpoint (see [Basu and Christensen, 2013; Singla *et al.*, 2013]).

2 Model and Preliminaries

The decision maker has access to n arms denoted $\{1, 2, ..., n\}$. At each time step t ($t = 1, 2, ..., T$) he has to decide which arm to pull. Every time the decision maker pulls an arm, he collects a reward and his goal is to pull the arms in such a way that his policy regret (to be formally defined shortly) is minimized. We assume the decision maker knows the time horizon T in advance.

We model the reward process of the arms as follows: Each arm k has a fixed underlying reward function denoted by $f_k(.)$. When the decision maker pulls arm k for the mth time ($m \geq 1$), he receives an instantaneous reward equal to $f_k(m)$. The cumulative reward from pulling arm k for m times is denoted by $F_k(m)$ and is equal to $f_k(1) + f_k(2) + ... + f_k(m)$. We assume that all the reward functions are bounded from below by 0 and from above by 1[3].

A *deterministic* policy π of length T is a sequence of mappings $\langle \pi_1, \pi_2, ..., \pi_T \rangle$ from the histories to the arms. That is,

$$\pi_t : \{1, 2, ..., n\}^{t-1} \times [0, 1]^{t-1} \longrightarrow \{1, 2, ..., n\}$$

prescribes the arm that must be pulled at step t given the history of actions and rewards observed so far. Given the reward functions $f_1(.), ..., f_n(.)$ and a deterministic policy π, the arm that the policy picks at each time step t is deterministic and is denoted by i_t. Let $b_k^t(\pi)$ be the instantaneous reward of arm k under policy π. More precisely, suppose the decision maker has followed policy π up until time $(t-1)$. $b_k^t(\pi)$ denotes the reward of playing arm k at time t. (As it is usually clear from the context what policy we are referring to, for simplicity we drop π in our notation.) Note the difference between $f_k(t)$ and b_k^t: unlike $f_k(t)$, b_k^t depends on the history of actions taken so far, in particular, the number of times arm k has been pulled before time t by policy π. We denote by $r(\pi)$ the total reward that a policy π collects, so $r(\pi) = \sum_{t=1}^{T} b_{i_t}^t$. Note that the total reward of a policy only depends on the number of times each arm is pulled and not the order.

Given the reward functions $f_1(.), ..., f_n(.)$ and T, let OPT be the policy that maximizes the total reward. In the online setting, the decision maker does not know the reward functions in advance and seeks to design a (possibly randomized) algorithm $\mathcal{A}$ so that the total reward he collects is as close as possible to that of OPT. In other words, the decision maker's goal is to minimize his *policy regret* which is defined as $r(\text{OPT}) - \mathbb{E}r(\mathcal{A})$. We say that an online algorithm $\mathcal{A}$ has sublinear policy regret if

$$\lim_{T \to \infty} \frac{r(\text{OPT}) - \mathbb{E}r(\mathcal{A})}{T} = 0.$$

[3]It is easy to see that without this assumption no algorithm can guarantee sublinear policy regret.

3 Increasing Reward Functions

3.1 The Offline Setting

We first show that when the reward curves are all increasing, there exists a single best arm that the optimal policy must repeatedly pull. Despite this fact, merely having the guarantee of no external regret would not be sufficient to guarantee no policy regret here (see Example 1).

Proposition 1 *Suppose for all $1 \leq k \leq n$, $f_k(.)$ is increasing. Then there exists an arm k_T^* such that the optimal offline policy OPT consists of pulling k_T^* for all T trials.*

Proof Assume OPT pulls arm k, T_k times (so $\sum_{i=1}^{n} T_i = T$). Suppose there exist arms i, j for which $T_i, T_j > 0$. We claim that $f_i(0) = f_j(0) = f_i(T_i) = f_j(T_j)$. In other words, f_i, f_j are both flat and identical. If this holds, the policy OPT remains optimal if we replace every occurrence of i in it with j. Repeating this for any two arms i', j' with $T_{i'}, T_{j'} > 0$, we see that OPT consists of pulling a single arm T times.

To prove the above claim, we first note that since all the f_ks are increasing, it must be that $f_i(T_i) \geq f_j(T_j)$. Otherwise one could collect a reward larger than OPT by never pulling arm i and instead pulling arm j for $T_j + T_i$ times. But this is a contradiction with optimality of OPT. Similarly $f_i(T_i) \leq f_j(T_j)$ and therefore, we can conclude $f_i(T_i) = f_j(T_j)$. Next we observe that $f_i(0) \geq f_j(T_j)$. Otherwise, given that $f_i(T_i) \leq f_j(T_j)$, one could collect a reward larger than OPT by never pulling arm i and instead pulling arm j for T_i additional times. Combining this with the previous equality, we can conclude that $f_i(T_i) \geq f_i(0) \geq f_j(T_j) = f_i(T_i)$ and therefore, $f_i(0) = f_i(T_i)$. Similar argument holds for j as well. Therefore, $f_i(T_i) = f_i(0) = f_j(T_j) = f_j(0)$. This proves our claim and finishes the proof. ∎

3.2 The Online Setting

In addition to being increasing and bounded, we assume the learning curves satisfy *decreasing marginal returns* (see Section 5 for a discussion of why this assumption is needed). More precisely, for any arm k we assume

$$\forall t \geq 1 : f_k(t+1) - f_k(t) \leq f_k(t) - f_k(t-1)$$

If we think of the reward curves as continuous functions, the concavity of f_k would give us decreasing marginal returns. Therefore we slightly abuse the terminology and refer to this property as "concavity". We emphasize that the concavity assumption is very natural and common in the context of human learning (see for example [Son and Sethi, 2006]) and in particular, concave learning curves have been shown to arise in various laboratory environments (see for example [Jovanovic and Nyarko, 1995; Anderson and Schooler, 1991]).

For the case where the reward functions are all increasing, bounded and concave, we introduce an online algorithm whose policy regret bound, as we shall prove, is sublinear and optimal. Our algorithm is motivated by the following observation: Suppose we initially pull arm k, t times and observe $f_k(1), f_k(2), ..., f_k(t)$. After this, given that f_k is increasing, we can be sure that for any $t < s \leq T$, $f_k(s) \geq f_k(t)$. In other words, the additional reward that can be collected from arm k in the remaining $(T-t)$ steps is minimized if f_k flattens at t. We define the *pessimistic estimate* of the total reward from arm k (denoted by $q_k^t(T)$) to be equal to the total reward of a function f_k' that is identical to f_k up to t, and then flattens out[4] (see Figure 1 (a)).

In addition to the above lower bound, concavity yields an upper bound on future payoffs. The additional reward that can be collected from arm k in the remaining $(T-t)$ steps is maximized if the function continues to grow linearly with rate $(f_k(t) - f_k(t-1))$. We define the *optimistic estimate* of the total reward from arm k (denoted by $p_k^t(T)$) to be equal to the total reward of a function f_k'' that is identical to f_k up to t, and then continues to grow linearly with rate $(f_k(t) - f_k(t-1))$ until it hits 1 [5](See Figure 1 (a)).

Since all the reward curves are increasing, by Proposition 1 we know that there exists a single best arm that is repeatedly pulled by the optimal policy. Therefore we seek to detect this arm. Our algorithm operates as follows: it maintains a set of candidate arms in which the best arm is guaranteed to lie. At each round, it pulls all the arms in the candidate set exactly once, and updates both optimistic and pessimistic estimates of the reward from these arms. If at some round the optimistic estimate of an arm k is less than or equal to the pessimistic estimate of another arm in the candidate set, then we are sure that k can not be the optimal arm, and therefore the algorithm eliminates it from the candidate set. See algorithm 1 for the details. We refer to this algorithm by $\mathcal{A}_1$.

For the coming theorem, we make use of a quantity that captures the time required to distinguish the optimal arm from the others. More precisely, we define $\tau(T) = \max_k \tau_k(T)$ where

$$\tau_k(T) = \arg\min_t \left\{ q_{k_T^*}^t(T) > p_k^t(T) \right\}$$

and k_T^* is the optimal arm for the time horizon T. Thus $\tau_k(T)$ specifies the smallest number of times we need to pull both arm k and k_T^* so that the optimistic estimate of the total reward from arm k falls behind the pessimistic estimate of the total reward from arm k_T^*.

We now prove that the policy regret of $\mathcal{A}_1$ is bounded by $n\tau(T)$, and show that this is optimal. Eventually in Theorem 2 we shall prove the regret of $\mathcal{A}_1$ is in fact sublinear.

Theorem 1 *Suppose for all $1 \leq k \leq n$, $f_k(.)$ is concave, increasing, bounded from below by 0 and from above by 1. Then*[6]

$$r(OPT) - r(\mathcal{A}_1) \leq n\tau(T).$$

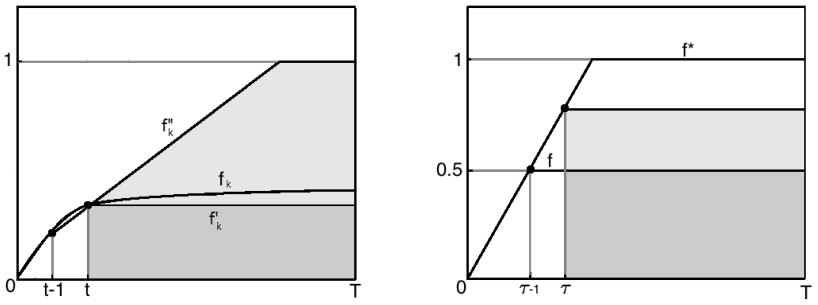
Furthermore, the policy regret bound of $\mathcal{A}_1$ is optimal: there exists a family of examples consisting of bounded, increasing and concave reward functions for which no algorithm can be guaranteed to have a policy regret bound of $o(n\tau(T))$.

Proof Observe that $\mathcal{A}_1$ detects the optimal arm after at most $\tau(T)$ phases. This is simply because a suboptimal arm k is

[4] To be more precise, $q_k^t(T) = F_k(t) + f_k(t)(T-t)$.

[5] To be more precise, $p_k^t(T) = F_k(t) + \sum_{s=t+1}^{T} \min\{1, f_k(t) + (f_k(t) - f_k(t-1))\,(s-t)\}$.

[6] One can easily see that with the exact same logic as the one in the proof of Theorem 1, we can get a slightly better upper bound of $\sum_{k=1}^{n} \tau_k(T)$ on the regret, but to simplify the statement of the theorem and its proof, we work with the worst case upper bound of $n\tau(T)$.



(a) (b)

Figure 1: (a) The optimistic (light grey) and pessimistic (dark grey) estimate. (b) The lower bound example.

Algorithm 1 The online algorithm for concave and increasing reward functions ($\mathcal{A}_1$)

$t = 0$; % t is the index of the current phase.
$S = \{1, ..., n\}$; % S is the set of remaining candidate arms.
Pull every arm exactly once.
repeat
  $t = t + 1$; % Start a new phase
  for each $i \in S$ **do**
    Pull arm i once.
    $p_i^t(T) = F_i(t) + \sum_{s=t+1}^{T} \min\{1, f_i(t) + (f(t) - f(t-1))(s - t)\}$; % Update the optimistic estimate.
    $q_i^t(T) = F_i(t) + f_i(t)(T - t)$; % Update the pessimistic estimate.
  end for
  for $i \neq j \in S$ **do**
    if $q_j^t(T) > p_i^t(T)$ % i.e. there exists an arm whose total reward is guaranteed to be larger than that of arm i **then**
      $S = S \setminus \{i\}$.
    end if
  end for
until only one arm remains in S or time runs out.
For the remaining steps (if any), pull the only arm left in S.

removed from S by phase $\tau_k(T)$. In addition, note that arm k_T^* can never be removed from S due to its optimality. Therefore, we can conclude that the number of times when $\mathcal{A}_1$ pulls suboptimals arm is at most $n\tau(T)$. Combining this with the fact that the reward per step is upper-bounded by 1 yields the desired inequality.

For the lower bound, fix a constant $\tau < T$, and consider n arms that all have linear reward curves with identical slope of $\frac{1}{2(\tau-1)}$, until they reach payoff 0.5 at time $(\tau - 1)$. Then $(n-1)$ of the curves stay at 0.5 forever, whereas one of them selected at random continues to 1 (see Figure 1 (b)). It is easy to see that for any $T > \tau$, $\tau(T) = \tau$. Using Yao's min-max principle [Yao, 1977], we lower-bound the regret of any deterministic algorithm on the above randomized setting. Let us define the number of arms an algorithm *verifies* to be equal to the number of arms that it pulls at least τ times (note that this is well-defined for any deterministic algorithm). It is easy to see that in this example, no algorithm can find the optimal arm with high probability by verifying $o(n)$ arms. To see this, first note that in order to see whether an arm is optimal, the algorithm has to pull it at least τ times. Assume, without loss of generality, that the first arm that the algorithm verifies is arm 1, the second one is arm 2, and so on. Since the index of the optimal arm is chosen uniformly at random, the expected number of arms the algorithm must verify before finding the optimal arm is equal to $\sum_{i=0}^{n-1} \frac{i}{n} = \frac{1}{n}\frac{n(n-1)}{2} = \frac{(n-1)}{2}$. Every time the algorithm verifies a sub-optimal arm, it fails to collect at least 0.5τ units of reward from the optimal arm, and instead obtains a reward on average equal to 0.25τ. As a result the policy regret of the algorithm is lower bounded by $\frac{\tau(n-2)}{2}$. ∎

Next we show that the policy regret of Algorithm 1 is in fact sublinear for any choice of bounded, concave and increasing reward functions $f_1(.), ..., f_n(.)$. The following notation will be useful in our argument: we denote the asymptote of $f_i(.)$ by a_i, i.e.

$$a_i = \lim_{t \to \infty} f_i(t).$$

Note that since $f_k(.)$ is increasing and bounded, the asymptote exists and is finite. Also for any arm i $\lim_{T\to\infty} \frac{F_i(T)}{T} = a_i$. We define a^* to be $\max_{1 \leq i \leq n} a_i$, and

$$\Delta_i(t) = f_i(t) - f_i(t-1).$$

Theorem 2 *For any set of bounded, concave, and increasing reward function $f_1(.), ..., f_n(.)$, the policy regret of $\mathcal{A}_1$ is sublinear.*

Here is the outline of the proof: We start by observing that for large values of T, the optimal arm must have the largest asymptote among other arms (note that we don't assume there is only one arm with maximum asymptote). Therefore a suboptimal arm i can be of one of the following two types:

1. The asymptote of arm i is smaller than that of the optimal arm; For this case, we show that $\mathcal{A}_1$ dismisses arm

i from its candidate set within $o(T)$ phases. As a result the regret from pulling arm i cannot not large.

2. The asymptote of arm i is equal to that of the optimal arm; For this case, we show that while $\mathcal{A}_1$ may fail to determine the suboptimality of arm i quickly, since the asymptote of i is as large as the optimal arm, pulling it does not add much to the policy regret of $\mathcal{A}_1$.

Proof Observe that $\lim_{T\to\infty} a_{k_T^*} = a^*$; in other words, if T is sufficiently large, then $k_T^* \in \arg\max_{1\le i\le n} a_i$. Throughout, we assume T is large enough so that the latter holds.

Let $W = \arg\max_k a_k$. We start by showing that if $i \notin W$ (i.e. $a_i < a^*$), then $\mathcal{A}_1$ removes arm i from its candidate set within $o(T)$ phases.

Lemma 1 *For arm i, if $a_i < a^*$, then $\lim_{T\to\infty} \frac{\tau_i(T)}{T} = 0$.*

Proof Suppose that the statement of the lemma does not hold and $\lim_{T\to\infty} \frac{\tau_i(T)}{T} \neq 0$. This means that for any unbounded and increasing sequence $\{\gamma_T\}_{T=1}^{\infty}$ for which $\lim_{T\to\infty} \frac{\gamma_T}{T} = 0$, there exists an infinite subsequence of T's such that $q_{k^*}^{\gamma_T}(T) < p_i^{\gamma_T}(T)$. Expanding this, we have[7]

$$\begin{aligned} F_{k_T^*}(\gamma) + (T-\gamma) f_{k_T^*}(\gamma) &\le F_i(\gamma) + \sum_{s>\gamma} \min\{1, f_i(\gamma) + \Delta_i(s-\gamma)\} \\ &\le F_i(\gamma) + \sum_{s>\gamma} f_i(\gamma) + \Delta_i(s-\gamma) \end{aligned}$$

The above is equivalent to

$$\begin{aligned} \left(F_{k_T^*}(\gamma) - F_i(\gamma)\right) + (T-\gamma)\left(f_{k_T^*}(\gamma) - f_i(\gamma)\right) &\le \sum_{s>\gamma} \Delta_i(s-\gamma) \\ &\le \Delta_i \frac{(T-\gamma+1)^2}{2} \\ &\le \Delta_i \frac{T^2}{2} \end{aligned}$$

Therefore combined we have:

$$F_{k_T^*}(\gamma_T) - F_i(\gamma_T) + (T-\gamma_T)\left(f_{k_T^*}(\gamma_T) - f_i(\gamma_T)\right) < \Delta_i(\gamma_T)\frac{T^2}{2} \quad (1)$$

Now note that if T (and as a result γ_T) is large enough, then $F_{k_T^*}(\gamma_T) \ge F_i(\gamma_T)$ and $f_{k_T^*}(\gamma_T) - f_i(\gamma_T) \ge \frac{a_{k_T^*} - a_i}{2} = \frac{a^* - a_i}{2}$. Let $C = \frac{a^* - a_i}{2}$ (recall that due to our assumption about arm i, $C > 0$). Therefore from (1) we obtain that $C(T-\gamma_T) < \Delta_i(\gamma_T)\frac{T^2}{2}$ and as a result

$$\begin{aligned} & C(T-\gamma) < \Delta_i(\gamma)\frac{T^2}{2} \\ \Rightarrow\quad & \frac{C(T-\gamma)}{T} < \Delta_i(\gamma)\frac{T}{2} \\ \Rightarrow\quad & \lim_{T\to\infty} \frac{C(T-\gamma)}{T} \le \lim_{T\to\infty} \Delta_i(\gamma)\frac{T}{2} \\ \Rightarrow\quad & C \le \lim_{T\to\infty} \Delta_i(\gamma)\frac{T}{2} \end{aligned}$$

where the last inequality follows from the fact that $\lim_{T\to\infty} \frac{\gamma_T}{T} = 0$. So we have

$$C \le \lim_{T\to\infty} \Delta_i(\gamma_T)\frac{T}{2} \quad (2)$$

[7] To simplify the notation in this equation we drop the subscript T. Also by δ_i we mean $\Delta_i(\gamma)$.

Next, one can easily verify that $\lim_{T\to\infty} \Delta_i(\gamma_T)T = 0$. To see this note that

$$\begin{aligned} & \sum_{t=1}^{\infty} \Delta_i(t) \le 1 \\ \Rightarrow\quad & \sum_{t=1}^{\infty} \Delta_i(t)T \le T \\ \Rightarrow\quad & \lim_{t\to\infty} \Delta_i(t)T = 0 \\ \Rightarrow\quad & \lim_{t\to\infty} \Delta_i(\gamma_T)T = 0 \end{aligned}$$

where the last inequality follows from the fact that $\{\gamma_T\}_{T=1}^{\infty}$ is positive, increasing and unbounded. This combined with (2) yields $C \le 0$, which is a contradiction. ∎

Using Lemma 1, we can conclude that for any arm $i \notin W$ our algorithm can distinguish i from k^* within $o(T)$ phases. In other words, after a prefix of at most $o(T)$ many phases, $\mathcal{A}_1$ eliminates every arm $i \notin W$.

Second we show that if $i \in W$ (i.e. $a_i = a^*$), while $\mathcal{A}_1$ may fail to detect the suboptimality of arm i quickly, pulling arm i does not add much to the policy regret of $\mathcal{A}_1$.

Lemma 2 *Let T_i denote the number of times $\mathcal{A}_1$ pulls arm $i \in W$. Then*

$$\lim_{T\to\infty} \frac{F_{k_T^*}(T) - \sum_{i\in W} F_i(T_i)}{T} = 0.$$

Proof If $i \in W$ gets eliminated within $o(T)$ phases, it can not cause the algorithm to suffer from a linear policy regret. Now consider a subset W' of W consisting of any suboptimal arm i for which it takes $\Theta(T)$ steps for the algorithm to eliminate i.

Given that the arms not in W' can all be detected in $T' = o(T)$ time steps, we have that $T - \sum_{i\in S'} T_i = T' = o(T)$. Let $w \in W'$ be the arm for which $\frac{F_i(T_i)}{T_i}$ is the smallest. We have

$$\begin{aligned} F_{k_T^*}(T) - \sum_{i\in W'} F_i(T_i) &= F_{k_T^*}(T) - \sum_{i\in W'} T_i \frac{F_i(T_i)}{T_i} \\ &< F_{k_T^*}(T) - \sum_{i\in W'} T_i \frac{F_w(T_w)}{T_w} \\ &= T\frac{F_{k_T^*}(T)}{T} - (T-T')\frac{F_w(T_w)}{T_w} \\ &= T\left(\frac{F_{k_T^*}(T)}{T} - \frac{F_w(T_w)}{T_w}\right) + T'\frac{F_w(T_w)}{T_w} \\ &\le T\left(\frac{F_{k_T^*}(T)}{T} - \frac{F_w(T_w)}{T_w}\right) + o(T) \end{aligned}$$

That is,

$$F_{k_T^*}(T) - \sum_{i\in W'} F_i(T_i) \le T\left(\frac{F_{k_T^*}(T)}{T} - \frac{F_w(T_w)}{T_w}\right) + o(T) \quad (3)$$

It only remains to show that

$$T\left(\frac{F_{k_T^*}(T)}{T} - \frac{F_w(T_w)}{T_w}\right) = o(T) \quad (4)$$

But this is certainly true as $\lim_{T\to\infty} \frac{F_{k_T^*}(T)}{T} = \lim_{T_w\to\infty} \frac{F_w(T_w)}{T_w} = a^*$. Combining (3), (4), we have the desired result. ∎

Combining Lemma 1 and 2 we obtain the sublinearity of the policy regret for $\mathcal{A}_1$. ∎

Finally, we remark that if the decision maker only gets to observe a corrupted version of the rewards and the corruption is bounded by some $\epsilon > 0$, then $\mathcal{A}_1$ is guaranteed to have a total reward at least equal to $r(\text{OPT}) - n\tau^{\epsilon}(T)$. $\tau^{\epsilon}(T)$ is the corrupted version of $\tau(T)$ in which the optimistic and pessimistic estimate from an arm are computed by taking the presence of noise into account [8] .

3.3 Simulations

So far we have established two upper bounds on the regret of $\mathcal{A}_1$: in Theorem 1 we gave an upper bound of $n\tau(T)$, and in Theorem 2 we showed that the regret of is always sublinear. In this section we empirically investigate the performance of $\mathcal{A}_1$ on several illustrative reward curves, and observe that $\tau(T)$ (and the regret of our algorithm) are typically *significantly* sublinear in T.

Throughout, for simplicity we set n to 2 and consider two arms $1, 2$ where the asymptote of f_1 is 1 and that of f_2 is 0.5. In Figure 2, we report the value of regret and τ versus $T = 500, ..., 30000$ for three different sets of examples.

In the first column of Figure 2, $f_1(t) = 1 - t^{-0.5}$ and $f_2(t) = 0.5 - 0.5t^{-\alpha}$ where $\alpha = 0.1, 0.5, 1, 5$. Each of these values corresponds to a different rate of increase for f_2. The larger α is, the faster f_2 converges to its asymptote. In this example for all values of α, τ increases slowly (sub-linearly) with T and as a result regret consistently decreases with T. The reason for the slow growth of τ is that f_1 converges to its asymptote very quickly; in addition, the rate with which f_2 increases, approaches 0 fast. This enables the algorithm to find and dismiss the suboptimal arms early on.

In the second column, $f_1(t) = \min\{1, \frac{t}{30000}\}$ and $f_2(t) = \min\{0.5, 0.5(\frac{t}{30000})^{\alpha}\}$ where $\alpha = 0.03, 0.1, 0.4, 1$. The smaller α is, the faster f_2 converges to its asymptote. Here when τ increases linearly or faster with T, the regret increases as well. Note that this does not contradict Theorem 2 as the theorem holds for large enough values of T only. Notice that for $\alpha = 0.03, 0.1, 0.4$, τ spikes at around $T = 10000$ and then drops. The reason for this behavior is that at that point, the optimal arm changes from arm 2 to arm 1.

Finally in the third column of Figure 2, $f_1(t) = 1 - t^{-0.1}$ and $f_2(t) = 0.5 - 0.5t^{-\alpha}$ where $\alpha = 0.1, 0.5, 1, 5$. It might come as a surprise that in this example, at the points when τ peaks, regret actually drops. Of course this does not contradict Theorem 1. While this theorem guarantees small regret when τ grows slowly with T, if τ increases linearly or faster, it does not necessarily imply that Algorithm 1 must suffer a large regret. For example, when $\alpha = 5$ and $T \leq 2500$, arm 1 is the sub-optimal arm, however, its reward is just slightly worse than that of the optimal arm (arm 2). Given that the rate of increase of the sub-optimal arm is sufficiently large, the algorithm fails to detect the optimal arm and instead keeps pulling the two an equal number of times. However, since the reward from the sub-optimal arm is getting closer to that of the optimal arm, the regret decreases. Note that this was not the case for the example in column 2.

[8] More precisely, $\tau^{\epsilon}(T) = \max_k \tau_k^{\epsilon}(T)$ and $\tau_k^{\epsilon}(T) = \arg\min_t\{q_{k_T^*}^{t,\epsilon}(T) > p_k^{t,\epsilon}(T)\}$ where $p_k^{t,\epsilon}(T) = F_k(t) + \sum_{s=t+1}^{T} \min\{1, f_k(t) + \epsilon + (f(t) - f(t-1) + 2\epsilon)(s-t)\}$ and $q_k^{t,\epsilon}(T) = F_k(t) + (f_k(t) - \epsilon)(T - t)$.

4 Decreasing Reward Functions

4.1 Optimal Offline Policy

When all the reward functions are decreasing, there exists a simple greedy approach that can compute the optimal offline policy. The proposed policy, $\mathcal{A}_0$, works as follows: At each time, pull the arm that results in the highest instantaneous reward. More precisely, if up to time step t, arm i has been pulled t_i times ($1 \leq i \leq n$), then pull an arm in $\arg\max_i f_i(t_i + 1)$.

Proposition 2 *Suppose for all $1 \leq k \leq n$, $f_k(.)$ is decreasing. Then $r(\mathcal{A}_0) = r(OPT)$.*

Proof We prove that $\mathcal{A}_0$ maximizes the total reward for any set of reward functions using induction on T. If $T = 1$, it is obvious that the optimal action is to pull the arm that results in the highest instantaneous reward; more precisely, in order to maximize the reward, one should pull arm k^* where $k^* \in \arg\max_k f_k(1)$ and this is exactly what $\mathcal{A}_0$ does. Suppose that our claim holds for $T \leq m$. Now consider the case of $T = m + 1$. Consider the policy OPT of length $(m + 1)$ that maximizes the total reward for the given set of reward functions $f_1, ..., f_n$. Let $k^* \in \arg\max_k f_k(1)$ be the arm that $\mathcal{A}_0$ initially pulls. We first show that, without loss of generality, we can assume OPT pulls arm k^* at least once. Suppose it does not. Consider the last action that OPT takes. Suppose it pulls arm k for the T_k^*'th times. Due to the fact that f_k is decreasing and due to the definition of k^*, we have $f_k(T_k) \leq f_k(1) \leq f_{k^*}(1)$. In other words if instead of k, in the last step OPT pulls k^*, the total reward it collects can only improve. Therefore without loss of generality we can assume OPT pulls arm k^* at least once.

Now let $g_{k^*}(t) = f_{k^*}(t + 1)$. Remove the first occurrence of k^* from OPT, and call the remaining policy π. Obviously, π, which is of length m, must maximize the total reward from arms with reward curves $f_1, ..., g_{k^*}, ..., f_n$, otherwise the total reward of OPT could be increased and this contradicts the optimality of OPT. Now applying the induction hypothesis, we know that $\mathcal{A}_0$ also maximizes the total reward for $f_1, ..., g_{k^*}, ..., f_n$. Therefore we can conclude that the total reward that $\mathcal{A}_0$ collects is equal to that of π. This finishes the proof. ∎

4.2 The Online Setting

As the above theorem shows, in the case of decreasing reward functions, there does not necessarily exist a single best arm that an online algorithm should track. Rather, the optimal algorithm needs to quickly react and switch to another arm when the reward from the current one drops.

We introduce a greedy online algorithm that can guarantee constant (and therefore sublinear) policy regret when all the reward curves are decreasing. The algorithm, which we call $\mathcal{A}_2$, does not need to know T in advance and works as follows: Intuitively, after the initial prefix of n actions, $\mathcal{A}_2$

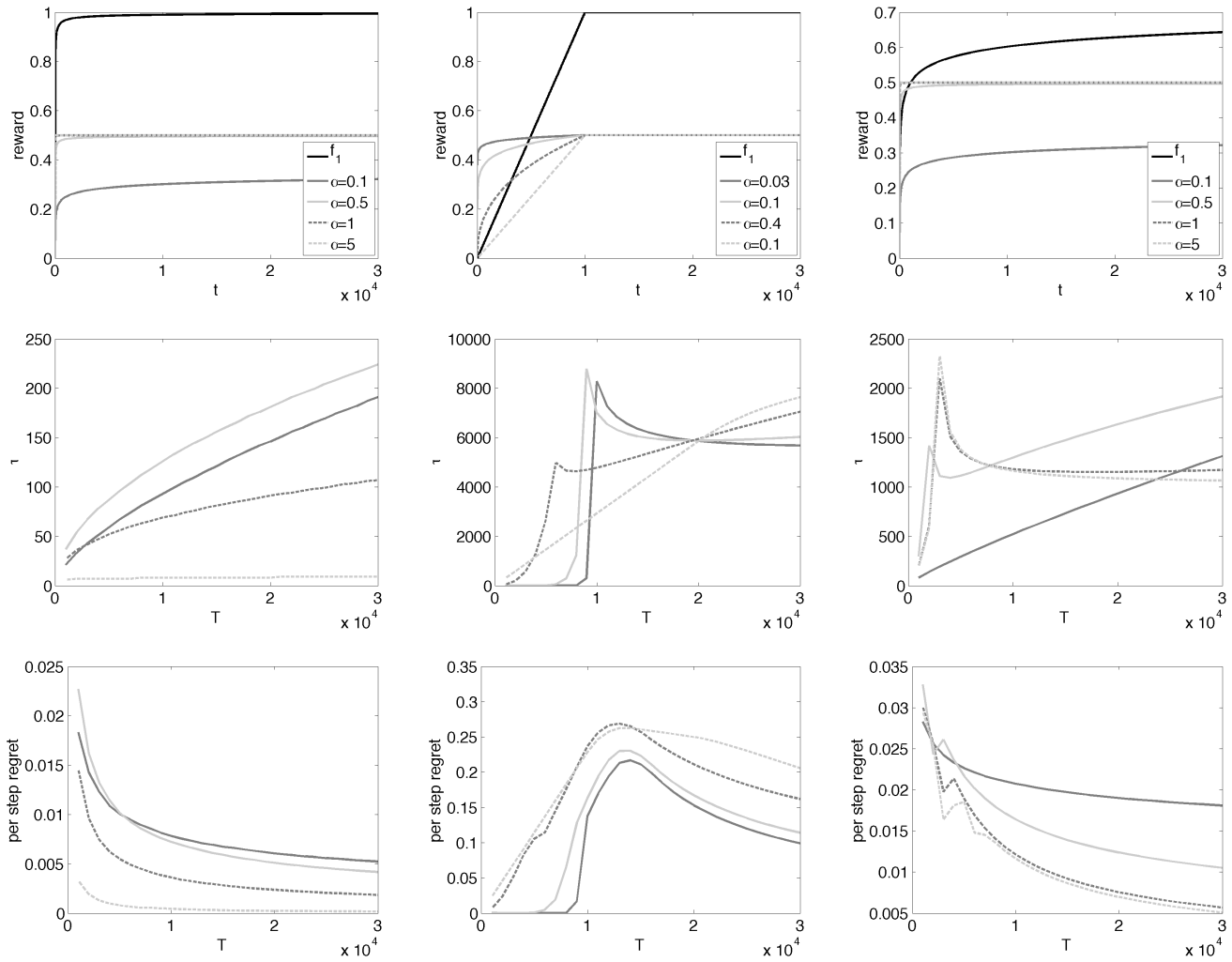


Figure 2: The first row illustrates f_1 (black) and f_2 (grey curves); the second and third rows illustrate τ and the regret respectively.

chooses a policy that is *identical* to the optimal greedy policy. The only difference is that for each action, it is getting a reward that is diminished by one additional time unit compared to the offline optimal benchmark. In sum, compared to the optimal algorithm, it loses at most one unit of reward from each action. But each instantaneous reward is bounded in $[0, 1]$, so the loss is at most n. See Algorithm 2 for further details.

Theorem 3 *Suppose for all $1 \leq k \leq n$, $f_k(.)$ is decreasing, bounded from below by 0 and from above by 1. Then*

$$r(OPT) - r(\mathcal{A}_2) \leq n.$$

Furthermore, the policy regret bound of $\mathcal{A}_2$ is optimal: there exists a family of examples consisting of bounded and decreasing reward functions for which no algorithm can be guaranteed to have a policy regret bound of $o(n)$.

Proof Suppose for all $1 \leq k \leq n$, OPT pulls arm k, T_k^* times and $\mathcal{A}_2$ pulls it T_k times. First note that for any arm k and any $t \leq \min\{T_k^*, T_k\}$, both OPT and $\mathcal{A}_2$ receive an instantaneous reward equal to $f_k(t)$ when they pull arm k for the t'th time. Thus these two instantaneous rewards cancel each other out in $r(\text{OPT}) - r(\mathcal{A}_2)$. It only remains to investigate the actions that $\mathcal{A}_2$ and OPT differ on.

Let U be the subset of arms for which $T_k > T_k^*$ and V be the subset of arms for which $T_k^* > T_k$. Note that if one of U or V is non-empty, the other one has to be non-empty as well. For any arm $i \in V$, let ℓ_i be the last instantaneous reward $\mathcal{A}_2$ collects from arm i, i.e. $\ell_i = f_i(T_i)$. Let $i^* = \arg\max_{i \in V} \ell_i$. Now consider the time when $\mathcal{A}_2$ pulls arm $j \in U$ for the tth time where $(T_j^* + 1) < t \leq T_j$ (if no such t exists, then $T_j^* + 1 = T_j$ which means $\mathcal{A}_2$ pulls arm j just once more than OPT). Suppose that so far, $\mathcal{A}_2$ has pulled arm i^*, $s \leq T_{i^*}$ times. Given that $\mathcal{A}_2$ chooses to pull arm j, we know that at that time $\mathcal{R}_j \geq \mathcal{R}_{i^*}$, or equivalently, $f_j(t-1) \geq f_{i^*}(s)$. Also, given that f_k's are all decreasing, we have that $f_{i^*}(s) \geq f_{i^*}(T_{i^*})$. Due to the way i^* has been chosen, we know that for all $i \in V$, $\ell_{i^*} \geq \ell_i$, or equivalently, $f_{i^*}(T_{i^*}) \geq$

Algorithm 2 A no policy regret algorithm for decreasing reward functions ($\mathcal{A}_2$)

for $1 \leq k \leq n$ **do**
  Pull arm k once
  $m_k = 1$; %m_k is the number of times arm k has been pulled so far
  $\mathcal{R}_k = f_k(1)$; %$\mathcal{R}_k$ is the most recent instantaneous reward collected from arm k
end for
for $n+1 \leq t \leq T$ **do**
  Pull arm j where $j \in \arg\max_k \mathcal{R}_k$
  $\mathcal{R}_j = f_j(m_j + 1)$
  $m_j = m_j + 1$
end for

$f_i(T_i)$. Given that f_k's are all decreasing, we have that for all $i \in V$ and $r \geq T_i$, $f_i(T_i) \geq f_i(r)$. Combining the last four inequalities, we have that for all $i \in V$ and $r \geq T_i$, $f_j(t-1) \geq f_i(r)$. This inequality means that except for at most 1 step (i.e. the T_jth time $\mathcal{A}_2$ pulls arm $j \in U$), those extra times at which $\mathcal{A}_2$ pulls arm j results in a reward larger than any of the instantaneous rewards that OPT collects and $\mathcal{A}_2$ doesn't. In other words $\mathcal{A}_2$ *wastes* at most 1 time step on arm j. Given that the rewards are all upper bounded by 1, we can conclude that $r(OPT) - r(\mathcal{A}_2) \leq |U| \leq n$.

For the lower bound, consider n arms, $(n-1)$ of which have a reward function equal $f(t) = \mathbf{1}[t = 1]$; and one of them (which we call k^*) chosen uniformly at random has a reward function always equal to 1. Note that in order to verify whether an arm is the optimal arm, any online algorithm has to pull it at least twice. If a suboptimal arm is pulled for the second time, we say the algorithm made a *mistake*. It is easy to see that in this example, no online algorithm can be guaranteed to find the optimal arm by making $o(n)$ mistakes in expectation. Assume without loss of generality that the first arm that the algorithm verifies (i.e. pulls for the second time) is arm 1, the second one is arm 2, and so on. Since the index of the optimal arm is chosen uniformly at random, the expected number of mistakes the algorithm makes before detecting k^* is equal to $\sum_{i=0}^{n-1} \frac{i}{n} = \frac{(n-1)}{2}$. This means that the algorithm makes $\Theta(n)$ mistakes in expectation. Given that every time the algorithm makes a mistake, it loses 1 unit of reward, the policy regret of the algorithm is lower bounded by $\frac{(n-1)}{2}$. This finishes the proof. ∎

We conclude with a few remarks: First, one can easily see that if the decision maker only gets to observe a corrupted version of the rewards and the magnitude of the corruption is bounded by $\epsilon > 0$, then $\mathcal{A}_2$ is guaranteed to have a total reward at least equal to $r(\text{OPT}) - n - \epsilon T$.

Second, we note that the greedy approach presented here fails to perform well in the increasing setting, since there the optimal arm can have the *lowest* payoff at time 1, and therefore never gets pulled by the greedy algorithm.

5 Future Directions

We presented no-policy regret algorithms for two settings: one in which all the reward functions were concave and increasing, and another where all the rewards were decreasing. We saw that even this simplified setting leads to non-trivial questions. We consider our work as a first step towards designing no-regret algorithms in settings where the rewards are neither stochastic nor fully adversarial.

Here are some interesting open questions raised by the results presented here.

- **Non-concave improving bandits.** The concavity assumption allows one to infer upper and lower bounds on the total reward of each arm with certainty. This is what our analysis relied on heavily to guarantee no policy regret. The same idea cannot be readily applied to settings where the rewards are increasing but non-concave. The existence of a no-policy regret algorithm for such settings remains an interesting open question.
- **Bandits that improve initially, then decay.** We assumed either all the reward functions are increasing or they are all decreasing. A natural and interesting question is whether it is possible to guarantee no policy regret in the case where the learning curves are concave and increasing at first, but then start decaying from some point on. It is easy to see that this is not simply achievable by naively combining the algorithms presented here.
- **Statistical noise.** Our algorithms can handle small amounts of adversarially generated noise. An important question is whether a similar regret bound can be shown to hold if the noise is generated stochastically. One natural idea to extend our work to this setting is to *estimate* the gradient of the reward functions and the corresponding confidence intervals using past observations, then follow a logic similar to what we proposed here.

References

[Anderson and Schooler, 1991] John R Anderson and Lael J Schooler. Reflections of the environment in memory. *Psychological science*, 2(6):396–408, 1991.

[Arora *et al.*, 2012] Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *International Conference on Machine Learning*, 2012.

[Atkinson *et al.*, 1965] Richard C Atkinson, Gordon H Bower, and Edward J Crothers. *Introduction to mathematical learning theory.* Wiley, 1965.

[Audibert and Bubeck, 2010] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, pages 13–p, 2010.

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[Barowy *et al.*, 2012] Daniel W Barowy, Charlie Curtsinger, Emery D Berger, and Andrew McGregor. Automan: a platform for integrating human-based and digital computation. In *Proceedings of the ACM international conference on Object oriented programming systems languages and applications*, 2012.

[Basu and Christensen, 2013] Sumit Basu and Janara Christensen. Teaching classification boundaries to humans. In *AAAI*, 2013.

[Chandrasekaran and Karp, 2012] Karthekeyan Chandrasekaran and Richard Karp. Finding a most biased coin with fewest flips. *arXiv preprint arXiv:1202.3639*, 2012.

[Du *et al.*, 2013] Shan Du, Mohammad Ibrahim, Mohamed Shehata, and Wael Badawy. Automatic license plate recognition (alpr): A state-of-the-art review. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(2):311–325, 2013.

[Gabillon *et al.*, 2013] Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems*, pages 2697–2705, 2013.

[Gittins *et al.*, 2011] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[Heidari and Kearns, 2013] Hoda Heidari and Michael Kearns. Depth-workload tradeoffs for workforce organization. In *AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[Jovanovic and Nyarko, 1995] Boyan Jovanovic and Yaw Nyarko. A bayesian learning model fitted to a variety of empirical learning curves. *Brookings Papers on Economic Activity. Microeconomics*, 1995:247–305, 1995.

[Mangal, 2009] SK Mangal. *An Introduction to Psychology*. Sterling Publishers Pvt. Ltd, 2009.

[Neu *et al.*, 2014] G. Neu, A. Gyorgy, Cs. Szepesvari, , and A. Antos. Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59, March 2014.

[Robbins, 1985] Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[Seung, 2015] Sebastian Seung. Eyewire: A game to map the brain. http://blog.eyewire.org/about/, 2015.

[Singla *et al.*, 2013] Adish Singla, Ilija Bogunovic, G Bartók, A Karbasi, and A Krause. On actively teaching the crowd to classify. 2013.

[Slivkins, 2011] Aleksandrs Slivkins. Monotone multi-armed bandit allocations. In *Conference on Learning Theory*, pages 829–834. Citeseer, 2011.

[Son and Sethi, 2006] Lisa K Son and Rajiv Sethi. Metacognitive control and optimal learning. *Cognitive Science*, 30(4):759–774, 2006.

[Streeter *et al.*, 2009] Matthew Streeter, Daniel Golovin, and Andreas Krause. Online learning of assignments. In *Advances in Neural Information Processing Systems*, pages 1794–1802, 2009.

[Tekin and Liu, 2012] Cem Tekin and Mingyan Liu. Online learning of rested and restless bandits. *Information Theory, IEEE Transactions on*, 58(8):5588–5611, 2012.

[Tran-Thanh *et al.*, 2014] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214:89–111, 2014.

[Yao, 1977] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science*, pages 222–227. IEEE, 1977.