Bandits with Costly Reward Observations

Aaron D. Tucker **Caleb Biddulph*** **Claire Wang*** **Thorsten Joachims**

Department of Computer Science, Cornell University, Ithaca NY USA
*Equal contribution, authors listed alphabetically

Abstract

Many machine learning applications rely on large datasets that are conveniently collected from existing sources or that are labeled automatically as a by-product of user actions. However, in settings such as content moderation, accurately and reliably labeled data comes at substantial cost. If a learning algorithm has to pay for reward information, for example by asking a human for feedback, how does this change the exploration/exploitation tradeoff? We study this question in the context of bandit learning. Specifically, we investigate Bandits with Costly Reward Observations, where a cost needs to be paid in order to observe the reward of the bandit's action. We show that the observation cost implies an $\Omega(c^{1/3}T^{2/3})$ lower bound on the regret. Furthermore, we develop a general non-adaptive bandit algorithm which matches this lower bound, and we present several competitive adaptive learning algorithms for both k-armed and contextual bandits.

1 INTRODUCTION

Machine learning has proven extremely successful on tasks where accurately labeled data is readily available and abundant, such as in speech recognition or online advertising. However, there are many crucial settings such as content moderation of ephemeral messages, where latency constraints force decisions to be made without human assessment, and yet obtaining accurate reward information necessarily involves some costly interaction which would not automatically happen otherwise. For example, while a search or ad engine can rely on users' clicks as a sufficiently accurate feedback signal, accurately labeling policy violations in content moderation is still based on human feedback. In these situations, there is a tradeoff between collecting more labels to achieve better performance and collecting fewer labels to avoid the labeling cost.

We first study this exploration/exploitation tradeoff in the k-armed bandit setting, then extend our results to a setting where the algorithm can decide that it needs additional oversight in some contexts but not in others. This problem is highly relevant to scalable oversight [Amodei et al., 2016], and more generally to modeling human preferences by learning from explicit human feedback [Hendrycks et al., 2021].

We refer to the specific setting studied in this paper as Bandits with Costly Reward Observations (BwCRO) (spoken bwick-roh). In this setting, the bandit problem is modified by adding a decision at each time step t to pay or not to pay a known cost c to observe the otherwise unknown reward r_t [Krueger et al., 2016]. As in standard bandit problems, a_t is the arm chosen at time t, and r_t depends on a_t. There are many different types of bandits that can be extended to the BwCRO setting, and their definitions are deferred to the relevant sections.

This setting can be used to analyze tradeoffs in a variety of domains. For example, an internet-of-things (IOT) device needs to account for its limited power supply during learning. Specifically, sensing the reward for an action can require substantial power, and the device needs to decide when to pay this updating cost. In chatbot optimization the chatbot needs to respond to questions in real-time, with an action space over possible utterances. However, observing the quality of an utterance can only be done through human assessment. Finally, in holistic recommendation we seek to maximize the value to a user according to a more holistic criterion than engagement metrics such as clicks. In this case, we would like to choose when it is worth it to get feedback from human assessors.

In this paper, we provide the following contributions. First, we prove an information-theoretic $\Omega(c^{1/3}T^{2/3})$ lower bound on the regret in the BwCRO setting. Second, we derive a novel algorithm for simple multi-armed BwCRO which provably matches these lower bounds up to a log-

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

arithmic factor. Third, we develop a general method for turning any suitable $O(T^{1/2})$-regret bandit algorithm into an $O(c^{1/3}T^{2/3})$-regret BwCRO algorithm. And, fourth, we propose a novel heuristic algorithm for linear contextual BwCRO which can adaptively choose when to query for a label depending on the context. Beyond the derivation of the new learning methods and their theoretical characterization, we also present experiments which validate and compare the empirical performance of the different algorithms.

2 SETTING DESCRIPTION

We first recap basic bandit settings and relevant related work, leading to a formal definition of the Bandits with Costly Reward Observations setting.

2.1 SETTING DESCRIPTION

Standard Bandit Settings. The multi-armed bandit setting creates a tradeoff between exploring new actions in order to understand their performance, and exploiting actions which have worked well in the past. There is a (sometimes null) set of contexts $\mathcal{X}$, set of actions $\mathcal{A}$ and an unknown mapping from actions to a distribution over rewards $\rho : \mathcal{X} \times \mathcal{A} \to \Delta(\mathbb{R})$. At each timestep t, the policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ chooses an action $a_t \in A_t \subset \mathcal{A}$ based on the context $x_t \in \mathcal{X}$, and receives a reward $r_t \sim \rho(x_t, a_t)$. However, the agent does not typically know ρ, and must learn how to choose high-reward actions over time. If we denote the optimal action at time t as $a_t^* = \arg\max_{a \in A_t} \mathbb{E}[r|X_t, a]$, then the agent seeks to minimize

$$\text{Regret} = \sum_{t=1}^{T} \left(\mathbb{E}[r|X_t, a_t^*] - \mathbb{E}[r|X_t, a_t] \right).$$

It is also sometimes useful to refer to the regret over an interval $[j, k]$, in which case we denote $\text{Regret}_{j:k} = \sum_{t=j}^{k} \mathbb{E}[r|X_t, a_t^*] - \sum_{t=j}^{k} \mathbb{E}[r|X_t, a_t]$.

Bandits with Costly Observations. The previous setting does not consider that observing labels may incur direct costs, rather than only having an opportunity cost. Bandits with costly observations (also studied as a special case of active reinforcement learning in Krueger et al. [2016]) adds an additional dimension that the algorithm must request a reward label and incur a known cost c in order to observe the reward. In order to add reward observation costs to a normal bandit setting, the contexts are unchanged and the new action space is $\mathcal{A}' = \mathcal{A} \times \{\text{label, no label}\}$. Define $l_t = 1$ if a label is requested and 0 otherwise, so that $a_t' = (a_t, l_t)$, and r_t is sampled as before, but the reward is modified to be $r_t' = r_t - c$ if a label is requested or $r_t' = r_t$ otherwise.

Since the algorithm always gets a higher immediate reward by not requesting a label, each requested label increases the regret by c. For the sake of clarity, if there are n labels we define the regret ignoring label costs as Regret°

$$\text{Regret}^\circ = \sum_{t=1}^{T} \left(\max_{a \in A_t} \mathbb{E}[r|X_t, a] - \mathbb{E}[r|X_t, a_t] \right),$$

and the regret including label cost as cRegret

$$\text{cRegret} = \text{Regret} = \text{Regret}^\circ + cn.$$

2.2 BACKGROUND AND RELATED WORK

While the Bandits with Costly Reward Observations setting has been studied before as a special case of different frameworks, our paper more thoroughly investigates this specific setting.

Active Learning. BwCRO is similar to active learning, but differs in that it pays a reward cost for additional labels rather than having a fixed labeling budget, and also makes labeling decisions one at a time in response to contextual information. There are several more closely related topics.

Partial Monitoring. BwCRO can be seen as a special case of partial monitoring, which studies sequential decision-making with imperfect feedback that may or may not include the reward. Prior work in partial monitoring states that since you need to take suboptimal actions in order for an algorithm to know if it is taking the optimal action, we will incur $O(T^{2/3})$ regret rate rather than an $O(T^{1/2})$ regret rate [Bartók et al., 2014]. However, our setting has a more specific structure where the cost between the observed and unobserved actions are exactly c, and we prove a novel $O(c^{1/3})$ component of the regret.

Best Arm Identification. BwCRO is related to best arm identification, which seeks to choose the arm with the highest expected reward in a multi-armed bandit setting with a fixed number of arms. All multi-armed bandit algorithms are related to best arm identification in that regret is incurred every time a suboptimal arm is chosen. Since the only way to get zero regret on a timestep is to choose the optimal arm, any algorithm with sublinear regret must eventually play the optimal arm the most often. Our regret lower bound proof uses the difficulty of identifying the optimal arm as a key component of the lower bound on regret, and our proposed Worth-it-Width algorithm can be seen as an ϵ-best arm identification algorithm. However, our other algorithms can achieve high performance in settings where the arms can change every timestep, which is a quite different setting.

Active Reinforcement Learning. BwCRO is closely related to active reinforcement learning, which adds a cost to observing the rewards in an RL setting. This work primarily focuses on MDPs instead of bandits [Krueger et al., 2016,

Schulze and Evans, 2018]. In contrast to this previous work, we develop new algorithms that not only have empirical advantages, but also proven regret rates. Furthermore, we prove the first lower bound for this setting.

3 ALGORITHMS

The core challenge of adding costly observations to bandit settings is that the algorithm must now decide when to request a label. We first present an algorithm which analyzes how to adapt the UCB algorithm to the BwCRO setting, then show more general algorithms which work in the BwCRO setting without the need for additional specialized analysis.

3.1 ALGORITHM FOR MULTI-ARMED BWCRO

For non-contextual bandits, the question of when to request labels can be simplified to the question of when to stop requesting labels. Consider any algorithm that decides whether or not to request a label based on a deterministic function $f(\mathcal{O}_t) \rightarrow \{\text{no label}, \text{label}\}$ of its observations $\mathcal{O}_t$ up to time t. Without contextual information, if the algorithm does not request a label at time t, then $\mathcal{O}_{t+1} = \mathcal{O}_t$. Therefore, $f(\mathcal{O}_t) = \text{no label}$ implies that $f(\mathcal{O}_{t+1}) = \text{no label}$, and once an algorithm stops requesting labels it will never request labels again. This means that many algorithms can be designed by focusing only on when to stop requesting labels, which forms the basis of Algorithms 1 and 2.

The key idea is to stop requesting labels by tracking whether or not it is still plausible that the bandit instance that it is observing is one where it will be worth it to collect enough data to disambiguate between the arms. If the difference between two arms' average rewards Δ is small enough, then it is cheaper to simply mistakenly commit to an arm than to pay for enough labels to figure out which arm is better. In the one-armed bandit case, it is possible to compute the range of Δs where it is better to request labels than to commit, and then check if the upper bound on Δ is such that it is still plausible that it is worth it to request labels. This idea can then be extended to the multi-armed case.

3.1.1 One-armed Bandit Setting and Algorithm

We first consider the simplified one-armed bandit setting where there is only one stochastic arm with unknown average reward. This allows us to analyze only the question of when to stop requesting labels compared to committing to a known alternative. In this setting, there are two options. The first is to choose an arm with a reward $r_t \in [0, 1]$ drawn stochastically from an unknown distribution with an unknown mean μ^*. The second is to choose a holdout arm with a known average reward ν. The per-step regret of choosing the wrong arm is $\Delta = |\mu^* - \nu|$.

Our goal is to decisively claim that one arm is better than another. If we know (with probability at least $1 - \delta$) that the stochastic or fixed arm is better, then there is no need for further labels. More formally, we define disambiguation.

Definition 3.1 (Disambiguate)**.** *Two arms a and a' with means μ^*_a and $\mu^*_{a'}$ are disambiguated if with probability at least $1 - \delta$ it can be said that either $\mu^*_a > \mu^*_{a'}$ or $\mu^*_a < \mu^*_{a'}$.*

How many labels does it take to disambiguate between the stochastic and the fixed arm? To help the analysis, define μ_n to be the empirical mean of the rewards of the stochastic arm based on n samples, and define $\hat{\Delta}_n = |\hat{\mu}_n - \nu|$.

Remark. *The stochastic and fixed arms are disambiguated after n stochastic arm labels if $n > 2\log(2T/\delta)/\hat{\Delta}_n^2$. For a fixed μ^*, this occurs by at most $n \leq 8\log(2T/\delta)/\Delta^2$.*

Proof. The Azuma-Hoeffding inequality [Azuma, 1967, Hoeffding, 1963] bounds the true average reward μ^* based on the observed rewards $\hat{\mu}_n$ with probability at least $1 - \delta$ for all timesteps [Agarwal et al., 2023]:

$$|\mu^* - \hat{\mu}_n| \leq \sqrt{2\log(2T/\delta)/n}$$

If $\mu^* \leq \nu$, then $\nu \leq u_n = \hat{\mu}_n + \sqrt{2\log(2T/\delta)/n}$ can only hold while $n \leq 2\log(2T/\delta)/\hat{\Delta}_n^2$. This further implies that $\mu^* \leq \nu$, since $\mu^* \leq u_n$ with probability $1 - \delta$. Similarly, if $\nu \leq \mu^*$, then $\hat{\mu}_n - \sqrt{2\log(2T/\delta)/n} = \ell_n \leq \nu$ can only hold while $n \leq 2\log(2T/\delta)/\hat{\Delta}_n^2$. Therefore, with high probability the two arms will be disambiguated once $n > 2\log(2T/\delta)/\hat{\Delta}_n^2$.

Bounding $\hat{\mu}_n$ with the Azuma-Hoeffding inequality $|\mu^* - \hat{\mu}_n| \leq \sqrt{2\log(2T/\delta)}$ shows that this will happen within at most $n < 8\log(2T/\delta)/\Delta^2$ steps. If $\nu < \mu^*$, then applying Azuma-Hoeffding we have $\mu^* - 2\sqrt{2\log(2T/\delta)/n} \leq \ell_n = \hat{\mu}_n - \sqrt{2\log(2T/\delta)/n}$, and so $\ell_n \leq \nu$ can only hold until $n > 8\log(2T/\delta)/\Delta^2$. Similarly, $\nu < \mu^*$, then applying Azuma-Hoeffding we have $\hat{\mu}_n + \sqrt{2\log(2T/\delta)/n} = u_n \leq \mu^* + 2\sqrt{2\log(2T/\delta)/n}$, and so $\ell_n \leq \nu$ can only hold until $n > 8\log(2T/\delta)/\Delta$. Therefore, with high probability the arms will be disambiguated by $n < 8\log(2T/\delta)/\Delta^2$. □

Note that the smaller the gap Δ between the fixed and stochastic arms, the less regret is accumulated by choosing the wrong arm. If Δ is small enough, then it is better to pay the regret of choosing the wrong arm than to pay the labeling cost needed to disambiguate between the two arms.

Remark. *Disambiguating between the stochastic and fixed arms is not worth it if the regret of choosing the wrong arm is lower than the labeling cost, which happens when*

$$\hat{\Delta}_n + \sqrt{2\log(2T/\delta)/n} < \sqrt[3]{8c\log(T/\delta)/T}.$$

Proof. Simply choosing an arm at the beginning yields an expected regret of $T\Delta$. The cost for requesting n labels is cn. For a given Δ, it takes $n < 8\log(2T/\delta)/\Delta^2$ labels until $\nu < \ell_n$ or $u_n > \nu$. This means that for a given Δ, the labeling cost to disambiguate arms is at most $8c\log(2T/\delta)/\Delta^2$.

If a Δ is always worth it to request enough labels that $\nu < \ell_n$ or $u_n > \nu$, then the maximum labeling cost must be less than the regret of simply choosing the wrong arm. Namely, it must satisfy $8c\log(T/\delta)/\Delta^2 \leq T\Delta$. Therefore it will not always be worth it to collect labels until either $\nu < \ell_n$ or $u_n > \nu$ if $\Delta < \sqrt[3]{8c\log(2T/\delta)/T}$.

However, the algorithm does not have access to $\Delta = |\mu^* - \nu|$. We can apply the triangle inequality and Azuma-Hoeffding inequality to bound Δ with probability $1-\delta$.

$$\Delta = |\nu - \mu| \leq |\nu - \hat{\mu}_n| + |\hat{\mu}_n - \mu| = \hat{\Delta}_n + \sqrt{2\log(2T/\delta)/n}.$$

Combining this bound on Δ and the definition of when it is always worth it to collect labels yields an expression usable by the algorithm: $\hat{\Delta}_n + \sqrt{2\log(2T/\delta)/n} < \sqrt[3]{8c\log(2T/\delta)/T}$. □

This tells us when to stop requesting labels – when the arms are disambiguated so we can confidently commit, or when an upper bound on Δ is small enough that committing to the wrong arm is cheaper than labeling until disambiguation.

3.1.2 Multi-armed Bandit Algorithm

In the multi-armed bandit setting there are multiple stochastic arms, and no arm has a known reward ν. However, the key idea from the single-armed case holds. If two arms a and a' have close enough expected values μ^a and $\mu^{a'}$, then it is better to pick one than to pay the cost of learning which is better. For this setting, the algorithm uses a time-dependent holdout reward ν_t which represents an expected reward that the algorithm can expect to get by committing now.

First, we set up notation to define ν_t. Define arm a's expected reward $\mu^a = \mathbb{E}[r_t | a_t = a]$, the empirical average at time t as $\hat{\mu}_t^a$, and n_t^a as the number of times arm a was observed at time t. Define u_t^a and ℓ_t^a as the Azuma-Hoeffding upper/lower bounds from $|\mu^a - \hat{\mu}_n^a| \leq \sqrt{2\log(2kT/\delta)/n_t^a}$, using the union bound to distribute the failure probability δ over all timesteps $t \in [T]$ and all k arms. Then if $\nu_t = \max_{a\in\mathcal{A}} \ell_t^a = \max_{a\in A}\left(\hat{\mu}_t^a - \sqrt{2\log(2kT/\delta)/n_t^a}\right)$ and $a_t^\nu = \arg\max_{a\in\mathcal{A}} \ell_t^a$, the algorithm can commit to the arm a_t^ν, and get at least reward ν_t. For convenience, denote the expected reward of arm a_t^ν as $\mu_t^\nu = \mu^{a_t^\nu}$.

Now, we extend the one-armed case by defining the stop conditions analogous to those of the one-armed case. Define the gaps g_t^a as $g_t^a = u_t^a - \nu_t$, which are upper bounds on the per-step regret for choosing the holdout arm a_t^ν instead of a, since with high probability both $\mu^a \leq u_t^a$ and $\nu_t \leq \mu^{a_t^\nu} = \mu_t^\nu$. Define the maximum gap $\bar{g}_t = \max_{a\in\mathcal{A}} g_t^a$ and the arm with the maximum gap $a_t^{\bar{g}} = \arg\max_{a\in\mathcal{A}} g_t^a$. Finally, define the worth-it-width

$$w = \sqrt[3]{8c\log(2kT/\delta)/T}.$$

Since $\bar{g}_t$ is an upper bound on the per-step regret of choosing the holdout arm a_t^ν once $\bar{g}_t \leq w$ committing to the holdout arm a_t^ν is better than gathering enough labels to conclude with high probability that some other arm a' has a higher average reward.

Algorithm 1 Worth-it-Width (WiW) Algorithm

At each time step t, compute the upper/lower bounds u_t^a & ℓ_t^a, holdout value ν_t, and max gap $\bar{g}_t$.
If $\bar{g}_t \leq w = \sqrt[3]{8c\log(2kT/\delta)/T}$, commit to arm a_t^ν.
Else if $g_t^{(a')} \leq w$ for all $a' \neq a_t^{\bar{g}}$ and the maximum gap $\bar{g}_t^a$ is such that $a_t^{\bar{g}} = a_t^\nu$, then commit to arm a_t^ν.
Otherwise, label arm $a = \arg\min_{a'\in\{a_t^{\bar{g}}, a_t^\nu\}} n_t^{(a')}$.

The next theorem establishes the regret of this algorithm.

Theorem 1 (Regret Rate for WiW Algorithm)**.** *Algorithm 1 has a regret rate of $\tilde{O}(kc^{1/3}T^{2/3})$ with high probability.*

Proof. Since it takes at most $8\log(2kT/\delta)/\Delta^2$ labels to disambiguate between two arms with a given Δ, we can play an arm at most $n = \sqrt[3]{8\log(2kT/\delta)(T/c)^{2/3}}$ times before concluding that μ^a or μ_t^ν is greater or that $|\mu^a - \mu_t^\nu| = \Delta \leq w$. Since we always play an arm associated with the largest gap we can only gather $k\sqrt[3]{8\log(2kT/\delta)(T/c)^{2/3}}$ labels before terminating, which incurs a regret of $k\sqrt[3]{8c\log(2kT/\delta)T^2}$.

Further, with high probability, g_t^a bounds the regret of committing to the holdout arm a_t^ν instead of arm a since $\mu^a - \mu^{a_t^\nu} \leq u_t^a - \ell_t^{a_t^\nu} = g_t^a$. At termination $g_t^a < \sqrt[3]{8c\log(2kT/\delta)/T}$, so our regret thereafter is bounded by $\sqrt[3]{8c\log(2kT/\delta)T^2}$ with high probability. Adding these two terms, the regret is $\tilde{O}(c^{1/3}T^{2/3})$. □

Algorithm 1 (WiW) directly exploits the insight that as arms have more and more similar expected rewards it gets harder to disambiguate between them while becoming cheaper to mistakenly commit. So, it commits if it disambiguates the arms or if the reward difference upper bound is less than w.

3.2 GENERAL ALGORITHM FOR BWCRO

The previous algorithm showed how to adapt to the UCB algorithm to the BwCRO setting, but required a detailed analysis of the algorithm to prove its regret rate. Is there a more general approach that does not require detailed analysis for each additional setting?

The affirmative answer is given by the following Fixed-N Algorithm, which is in fact very general and can also work in contextual bandit settings. Its key idea is to use a universally valid stopping criterion for requesting labels that is primarily a function of the horizon T and the label cost c.

Algorithm 2 Fixed-N Algorithm for Multi-armed Bandits

Given: Algorithm $\mathcal{A}$ that satisfies Assumption 3.1 with $\mathbb{E}\left[\text{Regret}^{\circ}_{1:n}\right] \leq K\sqrt{n}$,
Phase 1: Play according to $\mathcal{A}$ while observing the first $n = \left(\frac{TK}{2c}\right)^{2/3}$ labels.
Phase 2: Play according to $\mathcal{A}$ without more labels.

In order to analyze the performance of the algorithm, we first make an assumption that relates the regret of the algorithm after no longer requesting labels to its earlier performance.

Assumption 3.1 (Uniform Regret Rate)**.** *An algorithm $\mathcal{A}$ meets the uniform regret assumption if, for all $n \leq T$ and with randomness taken over the algorithm's choices and environment, a) playing according to $\mathcal{A}$ while observing labels for the first n timesteps results in $\mathbb{E}\left[\textit{Regret}^{\circ}_{1:n}\right] \in O(n^{1/2})$ and b) with randomness taken over the algorithm's choices and environment, and if requesting no further labels after the first n timesteps results in*

$$\frac{1}{T-n}\,\mathbb{E}\left[\textit{Regret}^{\circ}_{n+1:T}\right] \leq \frac{1}{n}\,\mathbb{E}\left[\textit{Regret}^{\circ}_{1:n}\right].$$

Part b of this assumption essentially states that average regret does not get worse with more labels. In particular, we can stop the algorithm after n labels at any time and expect an $O(n^{-1/2})$ per-timestep regret rate in retrospect and going forward. Part a states that the algorithm does not have any distinct phases that have qualitatively different regret evolutions. This excludes most explore-then-commit algorithms, but includes popular algorithms such as UCB and Thompson sampling. This assumption allows us to prove Theorem 2, which shows that Algorithm 2 achieves a regret rate of $O(c^{1/3}T^{2/3})$. If instead the base algorithm instead has an $\tilde{O}(T^{1/2})$ regret rate, then Algorithm 2 has the corresponding $\tilde{O}(c^{1/3}T^{2/3})$ regret rate.

Theorem 2 (Regret Rate for Fixed N Algorithm)**.** *Assuming that $\mathcal{A}$ satisfies the uniform regret assumption, the Fixed N algorithm based on $\mathcal{A}$ has cRegret $\in O(c^{1/3}T^{2/3})$.*

PROOF SKETCH. Assume that $\mathcal{A}$ satisfies the Uniform Regret assumption, so that $\mathbb{E}\left[\text{Regret}^{\circ}_{1:n}\right] \leq K\sqrt{n}$ for all $n > n_0$ for some n_0. In the BwCRO setting, receiving n labels incurs a regret of cn, so the total regret of using $\mathcal{A}$ while labeling the first $n > n_0$ can be bounded as follows:

$$\begin{aligned}
\text{cRegret}_{1:T} &= cn + \mathbb{E}\left[\text{Regret}^{\circ}_{1:n}\right] + \mathbb{E}\left[\text{Regret}^{\circ}_{n+1:T}\right] \\
&\leq cn + \frac{n}{n}\,\mathbb{E}\left[\text{Regret}^{\circ}_{1:n}\right] + \frac{T-n}{n}\,\mathbb{E}\left[\text{Regret}^{\circ}_{1:n}\right] \\
&= cn + T\,\mathbb{E}\left[\text{Regret}^{\circ}_{1:n}\right]/n \\
&\leq cn + TKn^{-1/2}
\end{aligned}$$

The first inequality follows from the Uniform Regret assumption, and the second from the definition of $O(\sqrt{n})$. As shown in Appendix A.3, $cn + TKn^{-1/2}$ is minimized by $n = (TK/2c)^{2/3}$. Plugging this value of n into the original bound $cn + TKn^{-1/2}$ yields the regret $O(c^{1/3}K^{2/3}T^{2/3}) \subset O(c^{1/3}T^{2/3})$. □

Algorithm 2 generalizes any bandit algorithm meeting Assumption 3.1 into a corresponding BwCRO algorithms. This provides a generic mechanism for constructing BwCRO algorithms, establishing a natural baseline for any special-purpose designed BwCRO algorithms.

3.3 LINEAR CONTEXTUAL BWCRO

While the Fixed-N algorithm is very general and can be used to handle costly reward observations in many bandit settings, it is entirely non-adaptive and does not use fewer labels in easier instances, or request labels in more interesting states. We conjecture that this is a substantial miss in many applications (such as healthcare, self-driving cars), where it is useful to request labels or oversight in the right states. The Worth-it-Width Algorithm (Algorithm 1) is adaptive, but has no notion of context or state. Is there a method for designing BwCRO algorithms that is both adaptive and general?

We combine the idea of keeping track of upper bounds on the per-step regret, and the idea of requesting labels only when it is worth it, to propose the following Δ Max Regret Heuristic for general BwCRO learning.

3.3.1 Δ Max Regret Heuristic

We can reinterpret the proof of the Fixed-N Algorithm regret rate (Theorem 2) to arrive at a more general algorithm. The proof places an upper bound on $\mathbb{E}\left[\text{cRegret}_{1:T}\right]$ as $cn + TKn^{-1/2}$, then minimizes that lower bound by selecting $n = (TK/2c)^{2/3}$ labels. However, we can interpret these mechanics instead as upper bounding the future $\mathbb{E}\left[\text{Regret}^{\circ}_{t:T}\right]$ as $TKn^{-1/2}$ (by Assumption 3.1), then requesting labels as long as the marginal labeling cost c exceeds the marginal decrease on our $\text{Regret}^{\circ}_{t:T}$ upper bound.

This suggests a new heuristic: request a label if the decrease in an upper bound on $\mathbb{E}\left[\text{Regret}^{\circ}_{t:T}\right]$ is greater than the labeling cost c. Denote the observation at time t as o_t, such that $\mathcal{O}_{t+1} = \mathcal{O}_t \cup \{o_t\}$ if a label is requested and $\mathcal{O}_{t+1} = \mathcal{O}_t$ otherwise, and let $\Phi(\mathcal{O}_t)$ be an upper bound on the per-step

Regret$^\circ$ given the observations $\mathcal{O}_t$. Then, request a label if

$$c \leq (T-t)\Phi(\mathcal{O}_t) - (T-t)\Phi(\mathcal{O}_t \cup \{o_t\}). \quad (1)$$

The Fixed-N algorithm can be exactly fit into this heuristic, while the Worth-it-Width algorithm can be seen as a refinement which stops slightly sooner. To recover the Fixed-N algorithm, then note that if $\mathcal{A}$ satisfies the Uniform Regret assumption, then we know that there exists a K such that

$$\frac{\mathbb{E}\left[\text{Regret}^\circ_{n+1:T}\right]}{T-n} \leq \frac{\mathbb{E}\left[\text{Regret}^\circ_{1:n}\right]}{n} \leq \frac{K\sqrt{n}}{n} = \frac{K}{\sqrt{n}},$$

and therefore there exists a per-step Regret$^\circ$ upper bound $\Phi(\mathcal{O}_t) = K/\sqrt{n}$. We loosen this bound to $TK/(\sqrt{n}(T-t))$ then bound $(T-t)\Phi(\mathcal{O}_t) - (T-t)\Phi(\mathcal{O}_t \cup \{o_t\})$ by $TK/(2\sqrt{n^3})$, then choose n such that $c = TK/\sqrt{n^3}$, which recovers the previous stopping condition of $n = (TK/2c)^{2/3}$.

This heuristic is more adaptive in two ways. First, by using an adaptive upper bound Φ (such as $\bar{g}_t$ in the Worth-it-Width Algorithm) instead of the non-adaptive upper bound $Kn^{-1/2}$ of the Uniform Regret Assumption, we can use instance-specific information to choose whether or not to label something. Second, this formulation allows us to take advantage of state-specific and not just instance-specific information in making our labeling decisions. We will concretely demonstrate this property by applying the heuristic to linear contextual bandits.

3.3.2 Linear Contextual BwCRO Algorithm

We adapt the Delta Max Regret heuristic to the linear contextual Bandits setting by building on top of LinUCB, a well-studied implementation of the "optimism in the face of uncertainty" principle [Li et al., 2010, Dani et al., 2008, Abbasi-yadkori et al., 2011]. In the linear contextual bandit setting, at each time step t the algorithm chooses an action from among k contexts $X_t = \{x_t^j \in \mathbb{R}^d\}_{j=1}^k$ which are drawn (at each time step) from some distribution $\mathcal{D}$ such that $\|x_t^j\| \leq B$. The algorithm receives reward $x_t \cdot \mu^* + \eta_t$ for the chosen $x_t \in X_t$, where μ^* is unknown, $\|\mu^*\| \leq W$, and η_t is σ^2 sub-Gaussian noise. Following Agarwal et al. [2023], define $\Sigma_t = (\sigma^2/W^2)I + \sum_{\tau=1}^{t-1} x_\tau x_\tau^T$, mean $\hat{\mu} = \Sigma_t^{-1}\sum_{\tau=1}^{t-1} r_\tau x_\tau$, $\beta_t = \sigma^2\left(2 + 4d\log(1 + tB^2W^2/d) + 8\log(4/\delta)\right)$, and an uncertainty region which contains the true μ^* at all time steps with probability $1-\delta$

$$\text{Ball}_t = \{\mu | \left(\hat{\mu}_t - \mu\right)^T \Sigma_t^{-1}\left(\hat{\mu}_t - \mu\right) \leq \beta_t\}.$$

LinUCB bounds the difference between our upper bound on the value of some x and its true value with the width of the uncertainty region along x. For any $\mu \in \text{Ball}_t$,

$$|\mu \cdot x - \hat{\mu}_t \cdot x| \leq \text{width}(\beta_t, \Sigma_t, x) = \sqrt{\beta_t x^T \Sigma_t^{-1} x}.$$

Since $\mu^* \in \text{Ball}_t$ for all $t \leq T$ with probability $1-\delta$, it follows that we can upper bound the value of $\mu^* \cdot x$ as $\mu^* \cdot x \leq \hat{\mu}_t \cdot x + \text{width}(\beta_t, \Sigma_t, x)$.

We now need a suitable Φ for the Delta Max Regret heuristic. Agarwal et al. [2023] provides a short proof that the per-step regret of choosing x_t is bounded as $2\text{width}(\beta_t, \Sigma_t, x_t)$. Define $x_t^* = \max_{x \in X_t} \mu^* x$, and $\tilde{\mu} = \arg\max_{\mu \in \text{Ball}_t} \max_{x \in X_t} \mu \cdot x = \arg\max_{\mu \in \text{Ball}_t} \mu \cdot x_t$.

$$\begin{aligned}
\text{Regret}_t &= \mu^* \cdot x_t^* - \mu^* \cdot x_t \\
&\leq \tilde{\mu} \cdot x_t - \mu^* \cdot x_t \\
&= (\tilde{\mu} - \hat{\mu}_t) \cdot x_t + (\hat{\mu}_t - \mu^*) \cdot x_t \\
&\leq 2\text{width}(\beta_t, \Sigma_t, x_t)
\end{aligned}$$

However, the algorithm does not know what the future X_t will be, so this bound cannot be applied directly. We instead consider the *maximum* width possible for a given covariance matrix, since this bounds x_t for all possible future contexts X_t. Conveniently, since the width depends on β_t and Σ_t, and since $\Sigma_{t+1} = \Sigma_t + x_t x_t^T$ if a label is requested, we can compute $\Phi(\mathcal{O}_t \cup \{o_t\})$ using information which is known at decision time. The max width is computed as

$$\Phi(\mathcal{O}_t) = \text{mw}(\beta, \Sigma) = \max_{\text{eigenvectors}} B\text{width}(\beta, \Sigma, e_i).$$

Algorithm 3 Delta Max Regret for LinUCB Algorithm

At each time step t, compute the center $\hat{\mu}_t$, covariance Σ_t, and uncertainty region Ball_t.
Play arm $x_t = \arg\max_{x \in D} \max_{\mu \in \text{Ball}_t} \mu \cdot x$.
Request label if x_t is such that

$$(T-t)\left[\text{mw}\left(\beta_t, \Sigma_t^{-1}\right) - \text{mw}\left(\beta_{t+1}, (\Sigma_t + x_t x_t^T)^{-1}\right)\right] > c$$

Otherwise don't request label

This algorithm is able to determine which states are useful to label, and our empirical evaluation will demonstrate that it can do this effectively on both synthetic and real data.

4 LOWER BOUND

We have described several algorithms, but are they close to optimal? The following proves that the BwCRO setting has a regret lower bound of $\Omega(c^{1/3}T^{2/3})$, and that therefore the Fixed N (Algorithm 2) and WiW (Algorithm 1) algorithms match the lower bound on the regret, and no algorithms can achieve a better asymptotic rate. This proves a novel rate for the labeling cost c, as well as agreeing with the rate for T from Krueger et al. [2016] and Bartók et al. [2014]. Our information-theoretic proof is based on the regret lower bound proof by Slivkins [2019].

Theorem 3. *The Bandits with Costly Observations setting has a regret lower bound of* $\Omega(c^{1/3}T^{2/3})$.

The basic idea of the proof is that we randomize over K instances with different best arms k^*, then show that (on average) k^* would not be played that often in a base instance, and therefore cannot be played that often in instance k^*.

Proof of Theorem 3. Consider a setting which chooses uniformly at random from K different multi-armed bandit instances, each with K actions where a coin is flipped with reward 1 for heads and 0 for tails. Denote the index of the randomly selected instance as k^*. In each bandit instance k, coin k is biased with expected reward $(1+\epsilon)/2$, and all other $K-1$ coins are fair. Denote the probability of an event A in instance k as $\Pr_k(a)$.

Consider an additional hypothetical base instance 0 where all the coins are fair. Our setting never chooses this instance. Let Q_k^T denote the number of times coin k is played in T timesteps, and note that by linearity of expectation,

$$\sum_{k=1}^{K} \mathbb{E}_0\left[Q_k^T\right] = \mathbb{E}_0\left[\sum_{k=1}^{K} Q_k^T\right] = \mathbb{E}_0[T] = T. \tag{2}$$

How many times do we play k^* in instance 0? Let $J_T = \{k : \mathbb{E}_0\left[Q_k^T\right] \leq 3T/K\}$ be the set of coins that the algorithm is not expected to play more than $3T/K$ times during the T timesteps in instance 0. For each coin $k \in J_T$, $\mathbb{E}_0[Q_k^T] \leq 3T/K$ so by the Markov inequality

$$\text{If } k \in J_T, \text{ then } \Pr_0\left(Q_k^T \leq 6T/K\right) \geq 1/2. \tag{3}$$

Further, J_T must have at least $2K/3$ elements, since its complement $\overline{J_T}$ must have at most $K/3$ elements, because otherwise the sum of the expectations of Q_k^T would be greater than T, which contradicts Equation 2.

$$\sum_{k\in\overline{J_T}} \mathbb{E}_0\left[Q_k^T\right] > \sum_{k\in\overline{J_T}}^{K} \frac{3T}{K} = |\overline{J_T}|\frac{3T}{K}$$

Since the coin k^* is chosen uniformly at random, and since J_T has at least $2K/3$ coins in it, with the randomness over the setting's choice of k^*,

$$\Pr(k^* \in J_T) > 2/3. \tag{4}$$

Combining Equations 3 and 4, we can bound the probability that $Q_{k^*}^T \leq 6T/K$ in instance 0. Denoting the event $Q_{k^*}^T \leq 6T/K$ as $\mathcal{E}$,

$$\Pr_0(\mathcal{E}) = \Pr_0\left(\mathcal{E}\middle|k^* \in J_T\right)\Pr\left(k^* \in J_T\right) \geq \frac{1}{2}\frac{2}{3} = \frac{1}{3}.$$

What is the regret in instance k^*? The KL Bound Lemma (proof in Lemma 1, Appendix A.2), states that for any event A based n observed coin flips $|\Pr_0(A) - \Pr_{k^*}(A)| \leq \epsilon\sqrt{n}$.

We can bound $\Pr_{k^*}(\mathcal{E})$ as

$$\Pr_{k^*}(\mathcal{E}) \geq \Pr_0(\mathcal{E}) - \epsilon\sqrt{n} \geq \frac{1}{3} - \epsilon\sqrt{n}. \tag{5}$$

In instance k^*, arm k^* is the optimal choice with expected reward $(1+\epsilon)/2$ while all other arms have expected reward $1/2$. Therefore, every timestep that k^* is not chosen incurs an expected regret of $\epsilon/2$. Since $Q_{k^*}^T$ is the number of times that k^* is chosen, and since there are T timesteps,

$$\mathbb{E}_{k^*}\left[\text{Regret}^\circ|Q_{k^*}^T\right] = \epsilon(T - Q_{k^*}^T)/2.$$

Therefore, if $\mathcal{E}$ then the regret in instance k^*

$$\mathbb{E}_{k^*}\left[\text{Regret}^\circ|\mathcal{E}\right] \geq \frac{\epsilon(T-6T/K)}{2} = \frac{(K-6)T\epsilon}{2K}. \tag{6}$$

Conclusion. If we collect n labels with a labeling cost c, then the regret in instance k^* is

$$\begin{aligned}\mathbb{E}_{k^*}\left[\text{cRegret}\right] &= \mathbb{E}_{k^*}\left[\text{Regret}^\circ\right] + cn \\ &\geq \mathbb{E}_{k^*}\left[\text{Regret}^\circ|\mathcal{E}\right]\Pr_{k^*}(\mathcal{E}) + cn \\ &\geq \frac{(K-6)T\epsilon}{2K}\left(\frac{1}{3} - \epsilon\sqrt{n}\right) + cn,\end{aligned}$$

with the first inequality coming from $\mathbb{E}_{k^*}\left[\text{Regret}^\circ\right] = \mathbb{E}_{k^*}\left[\text{Regret}^\circ|\mathcal{E}\right]\Pr_{k^*}(\mathcal{E}) + \mathbb{E}_{k^*}\left[\text{Regret}^\circ|\overline{\mathcal{E}}\right]\Pr_{k^*}(\overline{\mathcal{E}})$, and the second coming from using Equations 6 and 5. Choosing $\epsilon = \sqrt[3]{c/T}$ for the setting, we have

$$\mathbb{E}_{k^*}[\text{cRegret}] \geq \frac{(K-6)T\sqrt[3]{c/T}}{2K}\left(\frac{1}{3} - \sqrt[3]{c/T}\sqrt{n}\right) + cn$$

If the algorithm minimizes this expression with respect to n using $\sqrt{n} = \frac{(k-6)}{4k}\sqrt[3]{T/c}$, we get a regret of

$$\mathbb{E}_{j^*}\left[\text{cRegret}_T\right] \geq \frac{(k-6)}{6k}\sqrt[3]{cT^2} - \frac{(k-6)^2}{16k^2}\sqrt[3]{cT^2},$$

for an $\Omega(c^{1/3}T^{2/3})$ regret lower bound, as desired. □

5 EXPERIMENTS

We first compare the Worth-it-Width (WiW) algorithm (Algorithm 1), the Fixed-N algorithm (Algorithm 2), and the Delta Max Regret (DMR) algorithm (Equation 1) to a prior baseline [1] from Krueger et al. [2016], and the Naive UCB algorithm on a variety of synthetic non-contextual BwCRO problems. Then, we demonstrate the performance of the DMR algorithm (Algorithm 3) compared to Fixed-N and Naive UCB on both real and synthetic contextual problems.

[1] Note that Schulze and Evans [2018] present another more recent algorithm, however its much higher computational costs limit its evaluation to bandits with up to 40 timesteps in the original paper. As such, we defer its discussion to Appendix A.1.1.

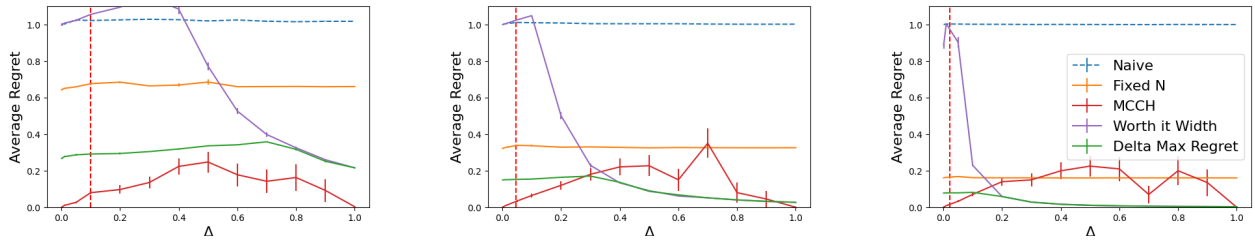

Figure 1: Final average per step regret for varying values of gaps Δ, $c = 1$, standard error from 20 trials. Dashed red line is at the predicted worst-case $\Delta = \sqrt[3]{c/T}$. Left graph has horizon $T = 1000$, middle has $T = 10000$, and right has $T = 100000$.

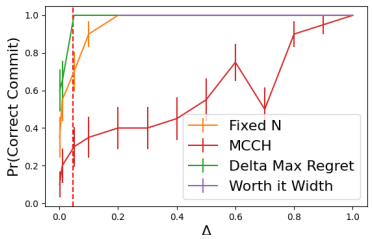

Figure 2: Probability of committing to the higher value arm for varying values of gaps Δ, $c = 1$, and $T = 10000$. Standard error from 20 trials. Dashed red line is at the predicted worst-case $\Delta = \sqrt[3]{c/T}$.

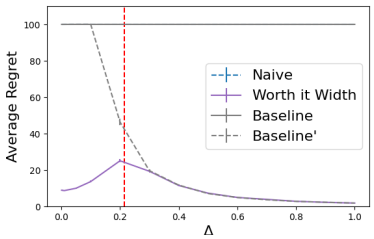

Figure 3: $c = 100, T = 10000$. The simpler baseline can achieve similar performance to the Worth-it-Width algorithm, but does considerably worse when the difference between arms Δ is small.

5.1 MULTI-ARMED BANDIT EVALUATION

We first consider a variety of two armed Bernoulli bandit settings with average reward $0.5 \pm \Delta/2$ and labeling cost $c = 1$. A more detailed study of cost is in Appendix A.1.4.

Baselines. We compare the performance of the WiW, Fixed N, and DMR Algorithms against two baselines. The first baseline is a naive implementation of UCB that always requests a label. The second baseline is the MCCH (Mind-changing Cost Heuristic) algorithm presented in Krueger et al. [2016], using UCB as the underlying algorithm, so that all algorithms are directly comparable. For Fixed N, the constant in the UCB algorithm is $K = 8\sqrt{k \log(Tk/\delta)}$.

Results. Figure 1 shows that the Fixed N algorithm consistently has low variance in its performance. Furthermore, its performance is invariant to different values of Δ, reflecting the fact that it does not adapt to problem instances at all. Figure 1 also shows that DMR consistently does at least as well as WiW, and shows substantial improvement for small Δ. Neither the MCCH nor the DMR algorithm dominates the other, with DMR having the advantage for larger Δ and longer episode lengths. As shown in Figure 2, this is a result of the fact that MCCH commits earlier, but is much more likely to commit to the wrong arm. For small arm differences Δ and short horizons T, this tradeoff is less costly. As Δ or T get bigger, the cost of wrongly committing is higher and the DMR and WiW algorithms do better.

Worth-it-Width Ablations. We also perform two ablations on the WiW algorithm to demonstrate the necessity of each step. “Baseline" simply plays UCB while requesting labels until one arm has a higher LCB than any other arm’s UCB. This baseline performs comparably to always requesting a label, since it never labels the suboptimal arm often enough to push its UCB below the optimal arm’s LCB. “Baseline’" rectifies this problem by playing and labeling the least played of the arms associated with the highest UCB and highest LCB, however it does not stop early if the associated gap is smaller than the "worth-it-width". This algorithm performs comparably to WiW for large differences Δ between the two arms but substantially worse for small differences, demonstrating the necessity of the early stopping condition. These results are corroborated over more parameter settings in Appendix A.1.5.

5.2 LINEAR CONTEXTUAL EVALUATION

We also evaluate the ability of the DMR algorithm to make state-dependent labeling decisions in contextual bandits.

Baselines. We compare the DMR algorithm to Fixed N and a Naive LinUCB baseline. The Naive LinUCB algorithm runs LinUCB and always requests a label. For the Fixed N Algorithm we choose $K = 8\beta_T d \log(1 + \frac{TB^2W^2}{d\sigma^2})$

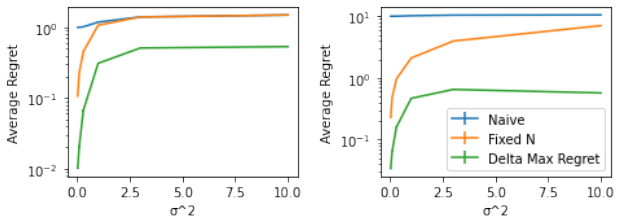

Figure 4: Final average per step regret for varying noises σ^2, standard error from 20 trials. Note the logarithmic y scale. The contexts have dimension $d = 5$, drawn from $\mathcal{N}(0, 1)^d$ and rescaled to size 1. Left has $c = 1$ and right has $c = 10$.

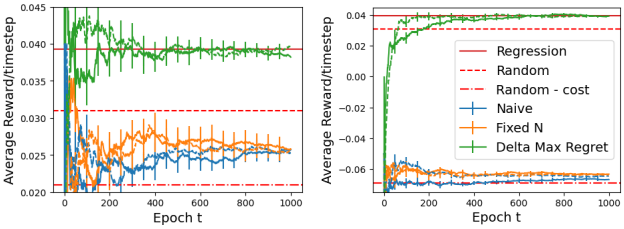

Figure 5: Average per-step reward at each timestep for 1000 timesteps rejection sampled using the Yahoo! Frontpage Dataset. Standard error from 20 trials. Left graph has cost $c = 0.01$ and right has $c = 0.1$. Non-red dashed lines correspond to using the doubling trick.

as per Agarwal et al. [2023].

Results. We set W and B (the sizes of μ^* and x) to 1, set $T = 10000$, and set the number of arms $k = 5$. As shown in Figure 4, increasing noise σ^2 forces all non-naive algorithms to request more labels, though DMR is able to avoid always requesting labels even with $\eta \sim \sqrt{10} * \mathcal{N}(0, 1)$ and each r_t constrained within $[-1, 1]$.

5.3 EVALUATION ON REAL-WORLD DATA

We also conducted experiments on the Yahoo! Front Page dataset [Chu et al., 2009] in order to validate the performance of the Delta Max Regret Algorithm on real data. This dataset was collected in an experiment where Yahoo! placed articles on their front page uniformly at random. Each context in X_t has a $D \times 5$ dimensional matrix with features for each of the D article available at the time, as well as a 5-dimensional vector representing information about the user. For convenience, we only use contexts which have exactly 20 articles. We create 35-dimensional vectors for each individual article by concatenating its 5-d vector to the user's 5-d vector, along with a 25-d vector containing the cross-terms of the user and article vectors. The fact that the articles were selected uniformly at random allows us to run an unbiased simulation of arbitrary policies using rejection sampling [Vanchinathan et al., 2014].

Baselines. In addition to the previously mentioned baselines, we also add a "Regression" skyline in order to understand the upper limits of performance for linear models in this setting. This regression model has the unfair advantage of being trained on all datapoints which could be sampled, and never paying a labeling cost. For any context, it then selects the action that has the highest predicted reward.

The Doubling Trick. In pratice, the horizon T may not be known beforehand. A standard method for handling this problem is to use the "Doubling Trick", where an initial T_0 is used as the horizon, and then the algorithm is rerun with $2T_0$ if the horizon is exceeded, then $4T_0$ if that horizon is exceeded, etc. We use a simpler variant where the horizon T is initially set to T_0, then modified in place by doubling it whenever it is exceeded. This reuses the old data rather than restarting, and simply updates the max-regret calculations of the remaining time and size of the confidence region.

Results. As seen in Figure 5, the DMR algorithm is able to achieve strong performance, doing as well as the regression model despite needing to request labels and pay the associated cost. In comparison, while the Fixed N algorithm is able to improve its performance over time, it performs poorly because it requests more labels than necessary. Further the doubling trick does not negatively impact the performance for any algorithm in the experiments, and in fact DMR seems to benefit in early timesteps. This shows that the doubling trick preserves not just asymptotic rate but also finite-sample performance in this setting, allowing the algorithms to easily adapt to unknown horizon lengths. The DMR algorithm is able to substantially save on labeling costs by successfully choosing informative contexts to label, while still attaining high performance and demonstrating the value of scalable oversight in linear contextual BwCRO.

6 CONCLUSIONS

We develop algorithms for Bandits with Costly Reward Observations, and provide theoretical guarantees on their regret. In particular, we develop the Fixed N algorithm for turning a large class of conventional bandit algorithms into algorithm for BwCRO, the WiW algorithm, and the DMR heuristic which can exploit instance-specific information in simple and contextual bandits. Finally, we prove $\Omega(c^{1/3}T^{2/3})$ lower bounds for BwCRO, matching the Fixed N regret rate.

Acknowledgements

This research was supported in part by NSF Awards IIS-1901168, IIS-2008139, and scholarship funding from Open Philanthropy. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/e1d5be1c7f2f456670de3d53c7b54f4a-Paper.pdf.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. *Reinforcement learning: Theory and algorithms*. 2023.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL https://arxiv.org/abs/1606.06565.

Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

Gábor Bartók, Dean P Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring—classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. A case study of behavior-driven conjoint analysis on Yahoo! Front Page Today module. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104, 2009.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. *Conference on Learning Theory*, page 355–366, 2008.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety, 2021. URL https://arxiv.org/abs/2109.13916.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

David Krueger, Jan Leike, Owain Evans, and John Salvatier. Active reinforcement learning: Observing rewards at a cost. *NeurIPS Future of Interactive Learning Machines (FILM) workshop*, 2016.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web - WWW '10*. ACM Press, 2010. doi: 10.1145/1772690.1772758. URL https://doi.org/10.1145%2F1772690.1772758.

Sebastian Schulze and Owain Evans. Active reinforcement learning with monte-carlo tree search. *CoRR*, abs/1803.04926, 2018. URL http://arxiv.org/abs/1803.04926.

Aleksandrs Slivkins. Introduction to multi-armed bandits, 2019. URL https://arxiv.org/abs/1904.07272.

Hastagiri P. Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, page 225–232, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450326681. doi: 10.1145/2645710.2645733. URL https://doi.org/10.1145/2645710.2645733.