Conservative Bandits

Yifan Wu YWU12@UALBERTA.CA
Roshan Shariff ROSHAN.SHARIFF@UALBERTA.CA
Tor Lattimore TOR.LATTIMORE@GMAIL.COM
Csaba Szepesvári SZEPESVA@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada

Abstract

We study a novel multi-armed bandit problem that models the challenge faced by a company wishing to explore new strategies to maximize revenue whilst simultaneously maintaining their revenue above a fixed baseline, uniformly over time. While previous work addressed the problem under the weaker requirement of maintaining the revenue constraint only at a given fixed time in the future, the design of those algorithms makes them unsuitable under the more stringent constraints. We consider both the stochastic and the adversarial settings, where we propose natural yet novel strategies and analyze the price for maintaining the constraints. Amongst other things, we prove both high probability and expectation bounds on the regret, while we also consider both the problem of maintaining the constraints with high probability or expectation. For the adversarial setting the price of maintaining the constraint appears to be higher, at least for the algorithm considered. A lower bound is given showing that the algorithm for the stochastic setting is almost optimal. Empirical results obtained in synthetic environments complement our theoretical findings.

1. Introduction

The manager of Zonlex, a fictional company, has just learned about bandit algorithms and is very excited about the opportunity to use this advanced technology to maximize Zonlex's revenue by optimizing the content on the landing page of the company's website. Every click on the content of their website pays a small reward; thanks to the high traffic that Zonlex's website enjoys, this translates into a decent revenue stream. Currently, Zonlex chooses the website's contents using a strategy designed over the years by its best engineers, but the manager suspects that some alternative strategies could potentially extract significantly more revenue. The manager is willing to explore bandit algorithms to identify the winning strategy. The manager's problem is that Zonlex cannot afford to lose more than 10% of its current revenue during its day-to-day operations and *at any given point in time*, as Zonlex needs a lot of cash to support its operations. The manager is aware that standard bandit algorithms experiment "wildly", at least initially, and as such may initially lose too much revenue and jeopardize the company's stable operations. As a result, the manager is afraid of deploying cutting-edge bandit methods, but notes that this just seems to be a chicken-and-egg problem: a learning algorithm cannot explore due to the potential high loss, whereas it must explore to be good in the long run.

The problem described in the previous paragraph is ubiquitous. It is present, for example, when attempting to learn better human-computer interaction strategies, say in dialogue systems or educational games. In these cases a designer may feel that experimenting with sub-par interaction strategies could cause more harm than good (e.g., Rieser and Lemon, 2008; Liu et al., 2014). Similarly, optimizing a production process in a factory via learning (and experimentation) has much potential (e.g., Gabel and Riedmiller, 2011), but deviating too much from established "best practices" will often be considered too dangerous. For examples from other domains see the survey paper of García and Fernández (2015).

Staying with Zonlex, the manager also knows that the standard practice in today's internet companies is to employ A/B testing on an appropriately small percentage of the traffic for some period of time (e.g., 10% in the case of Zonlex). The manager even thinks that perhaps a best-arm identification strategy from the bandit literature, such as the recent lil'UCB method of Jamieson et al. (2014), could be more suitable. While this is appealing, identifying the

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

best possible option may need too much time even with a good learning algorithm (e.g., this happens when the difference in payoff between the best and second best strategies is small). One can of course stop earlier, but then the potential for improvement is wasted: when to stop then becomes a delicate question on its own. As Zonlex only plans for the next five years anyway, they could adopt the more principled yet quite simple approach of first using their default favorite strategy until enough payoff is collected, so that in the time remaining of the five years the return-constraint is guaranteed to hold regardless of the future payoffs. While this is a solution, the manager suspects that other approaches may exist. One such potential approach is to discourage a given bandit algorithm from exploring the alternative options, while in some way encouraging its willingness to use the default option. In fact, this approach has been studied recently by Lattimore (2015a) (in a slightly more general setting than ours). However, the algorithm of Lattimore (2015a) cannot be guaranteed to maintain the return constraint *uniformly in time*. It is thus unsuitable for the conservative manager of Zonlex; a modification of the algorithm could possibly meet this stronger requirement, but it appears that this will substantially increase the worst-case regret.

In this paper we ask whether better approaches than the above naive one exist in the context of multi-armed bandits, and whether the existing approaches can achieve the best possible regret given the uniform constraint on the total return. In particular, our contributions are as follows: **(i)** Starting from multi-armed bandits, we first formulate what we call the family of "conservative bandit problems". As expected in these problems, the goal is to design learning algorithms that minimize regret under the additional constraint that at any given point in time, the total reward (return) must stay above a fixed percentage of the return of a fixed default arm, i.e., the return constraint must hold *uniformly in time*. The variants differ in terms of how stringent the constraint is (i.e., should the constraint hold in expectation, or with high probability?), whether the bandit problem is stochastic or adversarial, and whether the default arm's payoff is known before learning starts. **(ii)** We analyze the naive build-budget-then-learn strategy described above (which we call BudgetFirst) and design a significantly better alternative for stochastic bandits that switches between using the default arm and learning (using a version of UCB, a simple yet effective bandit learning algorithm: Agrawal, 1995; Katehakis and Robbins, 1995; Auer et al., 2002) in a "smoother" fashion. **(iii)** We prove that the new algorithm, which we call Conservative UCB, meets the uniform return constraint (in various senses), while it can achieve significantly less regret than BudgetFirst. In particular, while BudgetFirst is shown to pay a *multiplicative penalty* in the regret for maintaining the return constraint, Conservative UCB only pays an *additive penalty*. We provide both high probability and expectation bounds, consider both high probability and expectation constraints on the return, and also consider the case when the payoff of the default arm is initially unknown. **(iv)** We also prove a lower bound on the best regret given the constraint and as a result show that the additive penalty is unavoidable; thus Conservative UCB achieves the optimal regret in a worst-case sense. While Unbalanced MOSS of Lattimore (2015a), when specialized to our setting, also achieves the optimal regret (as follows from the analysis of Lattimore, 2015a), as mentioned earlier it does not maintain the constraint uniformly in time (it will explore too much at the beginning of time); it also relies heavily on the knowledge of the mean payoff of the default strategy. **(v)** We also consider the *adversarial setting* where we design an algorithm similar to Conservative UCB: the algorithm uses an underlying "base" adversarial bandit strategy when it finds that the return so far is sufficiently higher than the minimum required return. We prove that the resulting method indeed maintains the return constraint uniformly in time and we also prove a high-probability bound on its regret. We find, however, that the additive penalty in this case is higher than in the stochastic case. Here, the Exp3-γ algorithm of Lattimore (2015a) is an alternative, but again, this algorithm is not able to maintain the return constraint uniformly in time. **(vi)** The theoretical analysis is complemented by synthetic experiments on simple bandit problems whose purpose is to validate that the newly designed algorithm is reasonable and to show that the algorithms' behave as dictated by the theory developed. We also compare our method to Unbalanced MOSS to provide a perspective to see how much is lost due to maintaining the return constraint uniformly over time. We also identify future work. In particular, we expect our paper to inspire further works in related, more complex online learning problems, such as contextual bandits, or even reinforcement learning.

1.1. Previous Work

Our constraint is equivalent to a constraint on the regret to a default strategy, or in the language of prediction-with-expert-advice, or bandit literature, regret to a default action. In the full information, mostly studied in the adversarial setting, much work has been devoted to understanding the price of such constraints (Hutter and Poland, 2005; Even-Dar et al., 2008; Koolen, 2013; Sani et al., 2014, e.g.,). In particular, Koolen (2013) studies the Pareto frontier of regret vectors (which contains the non-dominated worst-case regret vectors of all algorithms). The main lesson of these works is that in the full information setting even a constant regret to a fixed default action can be maintained with essentially no increase in the regret to the best action. The situation quickly deteriorates in the bandit setting as shown by

Lattimore (2015a). This is perhaps unsurprising given that, as opposed to the full information setting, in the bandit setting one needs to actively explore to get improved estimates of the actions' payoffs. As mentioned earlier, Lattimore describes two learning algorithms relevant to our setting: In the stochastic setting we consider, Unbalanced MOSS (and its relative, Unbalanced UCB) are able to achieve a constant regret penalty while maintaining the return constraint while Exp3-γ achieves a much better regret as compared to our strategy for the adversarial setting. However, *neither of these algorithms maintain the return constraint uniformly in time*. Neither will the constraint hold with high probability. While Unbalanced UCB achieves problem-dependent bounds, it has the same issues as Unbalanced MOSS with maintaining the return constraint. Also, all these strategies rely heavily on knowing the payoff of the default action.

More broadly, the issue of staying safe while exploring has long been recognized in reinforcement learning (RL). García and Fernández (2015) provides a comprehensive survey of the relevant literature. Lack of space prevents us from including much of this review. However, the short summary is that while the issue has been considered to be important, no previous approach addresses the problem from a theoretical angle. Also, while it has been recognized that adding constraints on the return is one way to ensure safety, as far as we know, maintaining the constraints during learning (as opposed to imposing them as a way of restricting the set of feasible policies) has not been considered in this literature. Our work, while it considers a much simpler setting, suggest a novel approach to address the safe exploration problem in RL.

Another line of work considers safe exploration in the related context of optimization (Sui et al., 2015). However, the techniques and the problem setting (e.g., objective) in this work is substantially different from ours.

2. Conservative Multi-Armed Bandits

The multi-armed bandit problem is a sequential decision-making task in which a learning agent repeatedly chooses an action (called an *arm*) and receives a reward corresponding to that action. We assume there are $K+1$ arms and denote the arm chosen by the agent in round $t \in \{1, 2, \ldots\}$ by $I_t \in \{0, \ldots, K\}$. There is a reward $X_{t,i}$ associated with each arm i at each round t and the agent receives the reward corresponding to its chosen arm, X_{t,I_t}. The agent does not observe the other rewards $X_{t,j}$ $(j \neq I_t)$.

The learning performance of an agent over a time horizon n is usually measured by its *regret*, which is the difference between its reward and what it could have achieved by consistently choosing the single best arm in hindsight:

$$R_n = \max_{i \in \{0,\ldots,K\}} \sum_{t=1}^{n} X_{t,i} - X_{t,I_t}. \tag{1}$$

An agent is failing to learn unless its regret grows sublinearly: $R_n \in o(n)$; good agents achieve $R_n \in O(\sqrt{n})$ or even $R_n \in O(\log n)$.

We also use the notation $T_i(n) = \sum_{t=1}^{n} \mathbb{1}\{I_t = i\}$ for the number of times the agent chooses arm i in the first n time steps.

2.1. Conservative Exploration

Let arm 0 correspond to the conservative default action with the other arms $1, \ldots, K$ being the alternatives to be explored. We want to be able to choose some $\alpha > 0$ and constrain the learner to earn at least a $1-\alpha$ fraction of the reward from simply playing arm 0:

$$\sum_{s=1}^{t} X_{s,I_s} \geq (1-\alpha) \sum_{s=1}^{t} X_{s,0} \quad \text{for all } t \in \{1, \ldots, n\}. \tag{2}$$

For the introductory example above $\alpha = 0.1$, which corresponds to losing at most 10% of the revenue compared to the default website. It should be clear that small values of α force the learner to be highly conservative, whereas larger α correspond to a weaker constraint.

We introduce a quantity Z_n, called the *budget*, which quantifies how close the constraint (2) is to being violated:

$$Z_t = \sum_{s=1}^{t} X_{s,I_s} - (1-\alpha) X_{s,0}; \tag{3}$$

the constraint is satisfied if and only if $Z_t \geq 0$ for all $t \in \{1, \ldots, n\}$. Note that the constraints must hold uniformly in time.

Our objective is to design algorithms that minimize the regret (1) while simultaneously satisfying the constraint (2). In the following sections, we will consider two variants of multi-armed bandits: the stochastic setting in Section 3 and the adversarial setting in Section 4. In each case we will design algorithms that satisfy different versions of the constraint and give regret guarantees.

One may wonder: what if we only care about $Z_n \geq 0$ instead of $Z_t \geq 0$ for all t. Although our algorithms are designed for satisfying the anytime constraint on Z_t our lower bound, which is based on $Z_n \geq 0$ only, shows that in the stochastic setting we cannot improve the regret guarantee even if we only want to satisfy the overall constraint $Z_n \geq 0$.

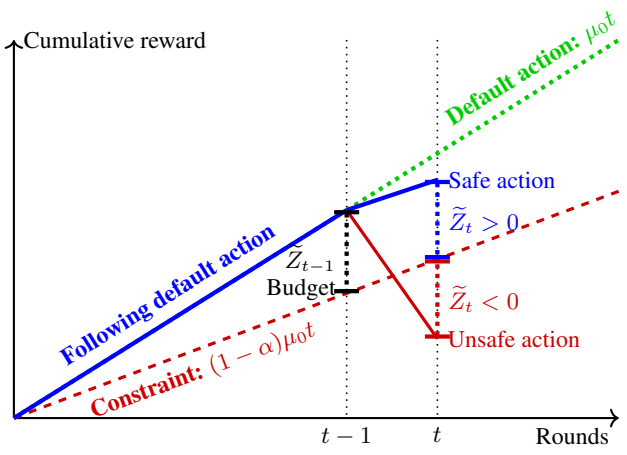

Figure 1. Choosing the default arm increases the budget. Then it is safe to explore a non-default arm if it cannot violate the constraint (i.e. make the budget negative).

3. The Stochastic Setting

In the stochastic multi-armed bandit setting each arm i and round t has a stochastic reward $X_{t,i} = \mu_i + \eta_{t,i}$, where $\mu_i \in [0,1]$ is the expected reward of arm i and the $\eta_{t,i}$ are independent random noise variables that we assume have 1-subgaussian distributions. We denote the expected reward of the optimal arm by $\mu^* = \max_i \mu_i$ and the gap between it and the expected reward of the ith arm by $\Delta_i = \mu^* - \mu_i$.

The regret R_n is now a random variable. We can bound it in expectation, of course, but we are often more interested in high-probability bounds on the weaker notion of *pseudo-regret:*

$$\widetilde{R}_n = n\mu^* - \sum_{t=1}^{n} \mu_{I_t} = \sum_{i=0}^{K} T_i(n)\Delta_i, \tag{4}$$

in which the noise in the arms' rewards is ignored and the randomness arises from the agent's choice of arm. The regret R_n and the pseudo-regret $\widetilde{R}_n$ are equal in expectation. High-probability bounds for the latter, however, can capture the risk of exploration without being dominated by the variance in the arms' rewards.

We use the notation $\hat{\mu}_i(n) = \frac{1}{T_i(n)} \sum_{t=1}^{n} \mathbb{1}\{I_t = i\} X_{t,i}$ for the empirical mean of the rewards from arm i observed by the agent in the first n rounds. If $T_i(n) = 0$ then we define $\hat{\mu}_i(n) = 0$. The algorithms for the stochastic setting will estimate the μ_i by $\hat{\mu}_i$ and will construct and act based on high-probability confidence intervals for the estimates.

3.1. The Budget Constraint

Just as we substituted regret with pseudo-regret, in the stochastic setting we will use the following form of the constraint (2):

$$\sum_{s=1}^{t} \mu_{I_s} \geq (1-\alpha)\mu_0 t \qquad \text{for all } t \in \{1,\dots,n\}; \tag{5}$$

the budget then becomes

$$\widetilde{Z}_t = \sum_{s=1}^{t} \mu_{I_s} - (1-\alpha)t\mu_0\,. \tag{6}$$

The default arm is always safe to play because it increases the budget by $\mu_0 - (1-\alpha)\mu_0 = \alpha\mu_0$. The budget will decrease for arms i with $\mu_i < (1-\alpha)\mu_0$; the constraint $\widetilde{Z}_n \geq 0$ is then in danger of being violated (Fig. 1).

In the following sections we will construct algorithms that satisfy pseudo-regret bounds and the budget constraint (5) with high probability $1-\delta$ (where $\delta > 0$ is a tunable parameter). In Section 3.4 we will see how these algorithms can be adapted to satisfy the constraint in expectation and with bounds on their expected regret.

For simplicity, we will initially assume that the algorithms know μ_0, the expected reward of the default arm. This is reasonable in situations where the default action has been used for a long time and is well-characterized. Even so, in Section 3.5 we will see that having to learn an unknown μ_0 is not a great hindrance.

3.2. BudgetFirst — A Naive Algorithm

Before presenting the new algorithm it is worth remarking on the most obvious naive attempt, which we call the BudgetFirst algorithm. A straightforward modification of UCB leads to an algorithm that accepts a confidence parameter $\delta \in (0,1)$ and suffers regret at most

$$\widetilde{R}_n = O\left(\sqrt{Kn\log\left(\frac{\log(n)}{\delta}\right)}\right) = R_{\text{worst}}\,. \tag{7}$$

Of course this algorithm alone will not satisfy the constraint (5), but that can be enforced by naively modifying the algorithm to deterministically choose $I_t = 0$ for the first t_0 rounds where

$$(\forall\, t_0 \leq t \leq n) \quad t\mu_0 - R_{\text{worst}} \geq (1-\alpha)t\mu_0\,.$$

Subsequently the algorithm plays the high probability version of UCB and the regret guarantee (7) ensures the constraint (5) is satisfied with high probability. Solving the equation above leads to $t_0 = \tilde{O}(R_{\text{worst}}/\alpha\mu_0)$, and since the regret while choosing the default arm may be $O(1)$ the worst-case regret guarantee of this approach is

$$\widetilde{R}_n = \Omega\left(\frac{1}{\mu_0\alpha}\sqrt{Kn\log\left(\frac{\log(n)}{\delta}\right)}\right)\,.$$

This is significantly worse than the more sophisticated algorithm that is our main contribution and for which the price of satisfying (5) is only an additive term rather than a large multiplicative factor.

3.3. Conservative UCB

A better strategy is to play the default arm only until the budget (6) is large enough to start exploring other arms with a low risk of violating the constraint. It is safe to keep exploring as long as the budget remains large, whereas if it decreases too much then it must be replenished by playing the default arm. In other words, we intersperse the exploration of a standard bandit algorithm with occasional budget-building phases when required. We show that accumulating a budget does not severely curtail exploration and thus gives small regret.

Conservative UCB (Algorithm 1) is based on UCB with the novel twist of maintaining a positive budget. In each round, UCB calculates upper confidence bounds for each arm; let J_t be the arm that maximizes this calculated confidence bound. Before playing this arm (as UCB would) our algorithm decides whether doing so risks the budget becoming negative. Of course, it does not know the actual budget $\widetilde{Z}_t$ because the μ_i ($i \neq 0$) are unknown; instead, it calculates a lower confidence bound ξ_t based on confidence intervals for the μ_i. More precisely, it calculates a lower confidence bound for what the budget would be if it played arm J_t. If this lower bound is positive then the constraint will not be violated as long as the confidence bounds hold. If so, the the algorithm chooses $I_t = J_t$ just as UCB would; otherwise it acts conservatively by choosing $I_t = 0$.

Remark 1 (Choosing ψ^δ)**.** The confidence intervals in Algorithm 1 are constructed using the function ψ^δ. Let F be the event that for all rounds $t \in \{1, 2, \ldots\}$ and every action $i \in [K]$, the confidence intervals are valid:

$$|\hat{\mu}_i(t) - \mu_i| \leq \sqrt{\frac{\psi^\delta(T_i(t))}{T_i(t)}}\,.$$

Our goal is to choose $\psi^\delta(\cdot)$ such that

$$\mathbb{P}\{F\} \geq 1 - \delta\,. \tag{8}$$

A simple choice is $\psi^\delta(s) = 2\log(Ks^3/\delta)$, for which (8) holds by Hoeffding's inequality and union bounds. The following choice achieve better performance in practice:

$$\psi^\delta(s) = \log\max\{3, \log\zeta\} + \log(2e^2\zeta) + \frac{\zeta(1+\log(\zeta))}{(\zeta-1)\log(\zeta)}\log\log(1+s), \tag{9}$$

where $\zeta = K/\delta$; it can be seen to achieve (8) by more careful analysis motivated by Garivier (2013),

Some remarks on Algorithm 1

1: **Input:** K, μ_0, δ, $\psi^\delta(\cdot)$
2: **for** $t \in 1, 2, \ldots$ **do**
▷ Compute confidence intervals...
3: $\theta_0(t), \lambda_0(t) \leftarrow \mu_0$ ▷ ...for known μ_0,
4: **for** $i \in 1, \ldots, K$ **do** ▷ ...for other arms,
5: $\Delta_i(t) \leftarrow \sqrt{\psi^\delta(T_i(t-1))/T_i(t-1)}$
6: $\theta_i(t) \leftarrow \hat{\mu}_i(t-1) + \Delta_i(t)$
7: $\lambda_i(t) \leftarrow \max\{0, \hat{\mu}_i(t-1) - \Delta_i(t)\}$
8: **end for**
9: $J_t \leftarrow \arg\max_i \theta_i(t)$ ▷ ...and find UCB arm.
▷ Compute budget and...
10: $\xi_t \leftarrow \sum_{s=1}^{t-1} \lambda_{I_s}(t) + \lambda_{J_t}(t) - (1-\alpha)t\mu_0$
11: **if** $\xi_t \geq 0$ **then**
12: $I_t \leftarrow J_t$ ▷ ...choose UCB arm if safe,
13: **else**
14: $I_t \leftarrow 0$ ▷ ...default arm otherwise.
15: **end if**
16: **end for**

Algorithm 1: Conservative UCB

- μ_0 is known, so the upper and lower confidence bounds can both be set to μ_0 (line 3). See Section 3.5 for a modification that learns an unknown μ_0.
- The max in the definition of the lower confidence bound $\lambda_i(t)$ (line 7) is because we have assumed $\mu_i \geq 0$ and so the lower confidence bound should never be less than 0.
- ξ_t (line 10) is a lower confidence bound on the budget (6) if action J_t is chosen. More precisely, it is a lower confidence bound on $\widetilde{Z}_t = \sum_{s=1}^{t-1} \mu_{I_s} + \mu_{J_t} - (1-\alpha)t\mu_0$.
- If the default arm is also the UCB arm ($J_t = 0$) and the confidence intervals all contain the true values, then $\mu^* = \mu_0$ and the algorithm will choose action 0 for all subsequent rounds, incurring no regret.

The following theorem guarantees that Conservative UCB satisfies the constraint while giving a high-probability upper bound on its regret.

Theorem 2. *In any stochastic environment where the arms have expected rewards $\mu_i \in [0,1]$ with 1-subgaussian noise, Algorithm 1 satisfies the following with probability at least $1-\delta$ and for every time horizon n, when ψ^δ is chosen in accordance with Remark 1 and with $L = \psi^\delta(n)$:*

$$\sum_{s=1}^{t} \mu_{I_s} \geq (1-\alpha)\mu_0 t \qquad \textit{for all } t \in \{1, \ldots, n\}, \tag{5}$$

$$\widetilde{R}_n \leq \sum_{i>0:\Delta_i>0} \left(\frac{4L}{\Delta_i} + \Delta_i\right) + \frac{2(K+1)\Delta_0}{\alpha\mu_0} + \frac{6L}{\alpha\mu_0}\sum_{i=1}^{K} \frac{\Delta_0}{\max\{\Delta_i, \Delta_0 - \Delta_i\}}, \quad (10)$$

$$\widetilde{R}_n \in O\left(\sqrt{nKL} + \frac{KL}{\alpha\mu_0}\right). \quad (11)$$

Standard unconstrained UCB algorithms achieve a regret of order $O(\sqrt{nKL})$; Theorem 2 tells us that the penalty our algorithm pays to satisfy the constraint is an extra additive regret of order $O(KL/\alpha\mu_0)$.

Remark 3. We take a moment to understand how the regret of the algorithm behaves if α is polynomial in $1/n$. Clearly if $\alpha \in O(1/n)$ then we have a constant exploration budget and the problem is trivially hard. In the slightly less extreme case when α is as small as n^{-a} for some $0 < a < 1$, the extra regret penalty is still not negligible: satisfying the constraint costs us $O(n^a)$ more regret in the worst case.

We would argue that the problem-dependent regret penalty (10) is more informative than the worst case of $O(n^a)$; our regret increases by

$$\frac{6L}{\alpha\mu_0}\sum_{i=1}^{K} \frac{\Delta_0}{\max\{\Delta_i, \Delta_0 - \Delta_i\}}.$$

Intuitively, even if α is very small, we can still explore as long as the default arm is close-to-optimal (i.e. Δ_0 is small) and most other arms are clearly sub-optimal (i.e. the Δ_i are large). Then the sub-optimal arms are quickly discarded and even the budget-building phases accrue little regret: the regret penalty remains quite small. More precisely, if $\Delta_0 \approx n^{-b_0}$ and $\min_{i>0:\Delta_i>0} \Delta_i \approx n^{-b}$, then the regret penalty is

$$O\left(n^{a+\min\{0, b-b_0\}}\right);$$

small Δ_0 and large Δ_i means $b - b_0 < 0$, giving a smaller penalty than the worst case of $O(n^a)$.

Remark 4. Curious readers may be wondering if $I_t = 0$ is the only conservative choice when the arm proposed by UCB risks violating the constraint. A natural alternative would be to use the lower confidence bound $\lambda_i(t)$ by choosing

$$I_t = \begin{cases} J_t\,, & \text{if } \xi_t \geq 0\,; \\ \arg\max_i \lambda_i(t)\,, & \text{otherwise}\,. \end{cases} \quad (12)$$

It is easy to see that if F does not occur, then choosing $\arg\max_i \lambda_i(t)$ increases the budget at least as much as choosing action 0 while incurring less regret and so this algorithm is preferable to Algorithm 1 in practice. Theoretically speaking, however, it is possible to show that the improvement is by at most a constant factor so our analysis of the simpler algorithm suffices. The proof of this claim is somewhat tedious so instead we provide two intuitions:

Firstly, the upper bound approximately matches the lower bound in the minimax regime, so any improvement must be relatively small in the minimax sense. Secondly, imagine we run the unmodified Algorithm 1 and let t be the first round when $I_t \neq J_t$ and where there exists an $i > 0$ with $\lambda_i(t) \geq \mu_0$. If F does not hold, then the actions chosen by UCB satisfy

$$T_i(t) \in \Omega\left(\min\left\{\frac{L}{\Delta_i^2}, \max_j T_j(t)\right\}\right),$$

which means that arms are being played in approximately the same frequency until they are proving suboptimal (for a similar proof, see Lattimore, 2015b). From this it follows that once $\lambda_{I_t}(t) \geq \mu_0$ for some i it will not be long before either $\lambda_j(t+s) \geq \mu_0$ or $T_j(t+s) \geq 4L/\Delta_i^2$ and in both cases the algorithm will cease playing conservatively. Thus it takes at most a constant proportion more time before the naive algorithm is exclusively choosing the arm chosen by UCB.

Next we discuss how small modifications to Algorithm 1 allow it to handle some variants of the problem while guaranteeing the same order of regret.

3.4. Considering the Expected Regret and Budget

One may care about the performance of the algorithm in expectation rather than with high probability, i.e. we want an upper bound on $\mathbb{E}\left[\widetilde{R}_n\right]$ and the constraint (5) becomes

$$\mathbb{E}\left[\sum_{s=1}^{t} \mu_{I_s}\right] \geq (1-\alpha)\mu_0 t, \quad \text{for all } t \in \{1, \ldots, n\}. \quad (13)$$

We argued in Remark 3 that if $\alpha \in O(1/n)$ then the problem is trivially hard; let us assume therefore that $\alpha \geq c/n$ for some $c > 1$. By running Algorithm 1 with $\delta = 1/n$ and $\alpha' = (\alpha - \delta)/(1 - \delta)$ we can achieve (13) and a regret bound with the same order as in Theorem 2.

To show (13) we have

$$\mathbb{E}\left[\sum_{s=1}^{t} \mu_{I_s}\right] \geq \mathbb{P}\{F\}\,\mathbb{E}\left[\sum_{s=1}^{t} \mu_{I_s} \,\middle|\, F\right] \geq (1-\delta)(1-\alpha')\mu_0 t = (1-\alpha)\mu_0 t\,.$$

As an upper bound, we have $\mathbb{E}[R_n] \leq \mathbb{E}[R_n|F] + \delta n = \mathbb{E}[R_n|F] + 1$. Here $\mathbb{E}[R_n|F]$ can be upper bounded by Theorem 2 with two changes: (i) L becomes $O(\log nK)$ after replacing δ with $1/n$, and (ii) α becomes α'. Since $\alpha'/\alpha \geq 1 - 1/c$ we get essentially the same order of regret bound as in Theorem 2.

3.5. Learning an Unknown μ_0

Two modifications to Algorithm 1 allow it to handle the case when μ_0 is unknown. First, just as we do for the non-

default arms, we need to set $\theta_0(t)$ and $\lambda_0(t)$ based on confidence intervals. Second, the lower bound on the budget needs to be set as

$$\xi'_t = \sum_{i=1}^{K} T_i(t-1)\lambda_i(t) + \lambda_{J_t}(t) + (T_0(t-1) - (1-\alpha)t)\theta_0(t)\,. \quad (14)$$

Theorem 5. *Algorithm 1, modified as above to work without knowing μ_0 but otherwise the same conditions as Theorem 2, satisfies with probability $1-\delta$ and for all time horizons n the constraint (5) and the regret bound*

$$\widetilde{R}_n \le \sum_{i:\Delta_i>0}\left(\frac{4L}{\Delta_i} + \Delta_i\right) + \frac{2(K+1)\Delta_0}{\alpha\mu_0} + \frac{7L}{\alpha\mu_0}\sum_{i=1}^{K}\frac{\Delta_0}{\max\{\Delta_i, \Delta_0 - \Delta_i\}}\,. \quad (15)$$

Theorem 5 shows that we get the same order of regret for unknown μ_0. The proof is very similar to the one for Theorem 2 and is also left for the appendix.

4. The Adversarial Setting

Unlike the stochastic case, in the adversarial multi-armed bandit setting we do not make any assumptions about how the rewards are generated. Instead, we analyze a learner's worst-case performance over all possible sequences of rewards $(X_{t,i})$. In effect, we are treating the environment as an adversary that has intimate knowledge of the learner's strategy and will devise a sequence of rewards that maximizes regret. To preserve some hope of succeeding, however, the learner is allowed to behave randomly: in each round it can randomize its choice of arm I_t using a distribution it constructs; the adversary cannot influence nor predict the result of this random choice.

Our goal is, as before, to satisfy the constraint (2) while bounding the regret (1) with high probability (the randomness comes from the learner's actions). We assume that the default arm has a fixed reward: $X_{t,0} = \mu_0 \in [0,1]$ for all t; the other arms' rewards are generated adversarially in $[0,1]$. The constraint to be satisfied then becomes $\sum_{s=1}^{t} X_{s,I_s} \ge (1-\alpha)\mu_0 t$ for all t.

Safe-playing strategy: We take any standard any-time high probability algorithm for adversarial bandits and adapt it to play as usual when it is safe to do so, i.e. when $Z_t \ge \sum_{s=1}^{t-1} X_{s,I_s} - (1-\alpha)\mu_0 t \ge 0$. Otherwise it should play $I_t = 0$. To demonstrate a regret bound, we only require that the bandit algorithm satisfy the following requirement.

Definition 6. An algorithm $\mathcal{A}$ is $\hat{R}_t^\delta$*-admissible* ($\hat{R}_t^\delta$ sub-linear) if for any δ, in the adversarial setting it satisfies

$$\mathbb{P}\left\{\forall t \in \{1,2,\ldots\}, R_t \le \hat{R}_t^\delta\right\} \ge 1-\delta.$$

Note that this performance requirement is stronger than the typical high probability bound but is nevertheless achievable. For example, Neu (2015) states the following for the any-time version of their algorithm: given any time horizon n and confidence level δ, $\mathbb{P}\left\{R_n \le \hat{R}'_n(\delta)\right\} \ge 1-\delta$ for some sub-linear $\hat{R}'_t(\delta)$. If we let $\hat{R}_t^\delta = \hat{R}'_t(\delta/2t^2)$ then $\mathbb{P}\left\{R_t \le \hat{R}_t^\delta\right\} \ge 1 - \frac{\delta}{2t^2}$ holds for any fixed t. Since the algorithm does not require n and δ as input, a union bound shows it to be $\hat{R}_t^\delta$-admissible.

Having satisfied ourselves that there are indeed algorithms that meet our requirements, we can prove a regret guarantee for our safe-playing strategy.

Theorem 7. *Any $\hat{R}_t^\delta$-admissible algorithm $\mathcal{A}$, when adapted with our safe-playing strategy, satisfies the constraint (2) and has a regret bound of $R_n \le t_0 + \hat{R}_n^\delta$ with probability at least $1-\delta$ where $t_0 = \max\{t \mid \alpha\mu_0 t \le \hat{R}_t^\delta + \mu_0\}$.*

Corollary 8. *The any-time high probability algorithm of Neu (2015) adapted with our safe-playing strategy gives $\hat{R}_t^\delta = 7\sqrt{Kt\log K}\log(4t^2/\delta)$ and*

$$R_n \le 7\sqrt{Kn\log K}\log(4n^2/\delta) + \frac{49K\log K}{\alpha^2\mu_0^2}\log^2\frac{4n^2}{\delta}$$

with probability at least $1-\delta$.

Corollary 8 shows that a strategy similar to that of Algorithm 1 also works for the adversarial setting. However, we pay a higher regret penalty to satisfy the constraint: $O\left(\frac{KL^2}{(\alpha\mu_0)^2}\right)$ rather than the $O\left(\frac{KL}{\alpha\mu_0}\right)$ we had in the stochastic setting. Whether this is because (i) our algorithm is sub-optimal, (ii) the analysis is not tight, or (iii) there is some intrinsic hardness in the non-stochastic setting is still not clear and remains an interesting open problem.

5. Lower Bound on the Regret

We now present a worst-case lower bound where α, μ_0 and n are fixed, but the mean rewards are free to change. For any vector $\mu \in [0,1]^K$, we will write $\mathbb{E}_\mu$ to denote expectations under the environment where all arms have normally-distributed unit-variance rewards and means μ_i (i.e., the fixed value μ_0 is the mean reward of arm 0 and the components of μ are the mean rewards of the other arms). We assume normally distributed noise for simplicity: other subgaussian distributions whose parameter is kept fixed independently of the mean rewards work identically.

Theorem 9. *Suppose for any $\mu_i \in [0,1]$ ($i > 0$) and μ_0*

satisfying

$$\min\{\mu_0, 1-\mu_0\} \geq \max\left\{1/2\sqrt{\alpha}, \sqrt{e+1/2}\right\}\sqrt{K/n},$$

an algorithm satisfies $\mathbb{E}_\mu \sum_{t=1}^n X_{t,I_t} \geq (1-\alpha)\mu_0 n$. *Then there is some* $\mu \in [0,1]^K$ *such that its expected regret satisfies* $\mathbb{E}_\mu R_n \geq B$ *where*

$$B = \max\left\{\frac{K}{(16e+8)\alpha\mu_0}, \frac{\sqrt{Kn}}{\sqrt{16e+8}}\right\}. \qquad (16)$$

Theorem 9 shows that our algorithm for the stochastic setting is near-optimal (up to a logarithmic factor L) in the worst case. A problem-dependent lower bound for the stochastic setting would be interesting but is left for future work. Also note that in the lower bound we only use $\mathbb{E}_\mu \sum_{t=1}^n X_t \geq (1-\alpha)n\mu_0$ for the last round n, which means that the regret guarantee cannot be improved if we only care about the last-round budget instead of the anytime budget. In practice, however, enforcing the constraint in all rounds will generally lead to significantly worse results because the algorithm cannot explore early on. This is demonstrated empirically in Section 6, where we find that the Unbalanced MOSS algorithm performs very well in terms of the expected regret, but does not satisfy the constraint in early rounds.

Remark 10. The theorem above almost follows from the lower bound given by Lattimore (2015a), but in that paper μ_0 is unknown, while here it may be known. This makes our result strictly stronger, as the lower bound is the same up to constant factors.

6. Experiments

We evaluate the performance of Conservative UCB compared to UCB and Unbalanced MOSS (Lattimore, 2015a) using simulated data in two regimes. In the first we fix the horizon and sweep over $\alpha \in [0,1]$ to show the degradation of the average regret of Conservative UCB relative to UCB as the constraint becomes harsher (α close to zero). In the second regime we fix $\alpha = 0.1$ and plot the long-term average regret, showing that Conservative UCB is eventually nearly as good as UCB, despite the constraint. Each data point is an average of $N \approx 4000$ i.i.d. samples, which makes error bars too small to see. Results are shown for both versions of Conservative UCB: The first knows the mean μ_0 of the default arm while the second does not and must act more conservatively while learning this value. As predicted by the theory, the difference in performance between these two versions of the algorithm is relatively small, but note that even when $\alpha = 1$ the algorithm that knows μ_0 is performing better because this knowledge is useful in the unconstrained setting. This is also true of the BudgetFirst algorithm, which is unconstrained when $\alpha = 1$ and exploits its knowledge of μ_0 to eliminate the default arm. This algorithm is so conservative that even when α is nearly zero it must first build a significant budget. We tuned the Unbalanced MOSS algorithm with the following parameters.

$$B_0 = \frac{nK}{\sqrt{nK} + \frac{K}{\alpha\mu_0}} \qquad B_i = B_K = \sqrt{nK} + \frac{K}{\alpha\mu_0}.$$

The quantity B_i determines the regret of the algorithm with respect to arm i up to constant factors, and must be chosen to lie inside the Pareto frontier given by Lattimore (2015a). It should be emphasised that Unbalanced MOSS does *not* constrain the return except for the last round, and has no high-probability guarantees. This freedom allows it to explore early, which gives it a significant advantage over the highly constrained Conservative UCB. Furthermore, it also requires $B_0, \ldots, B_K$ as inputs, which means that μ_0 must be known in advance. The mean rewards in both experiments are $\mu_0 = 0.5$, $\mu_1 = 0.6$, $\mu_2 = \mu_3 = \mu_4 = 0.4$, which means that the default arm is slightly sub-optimal.

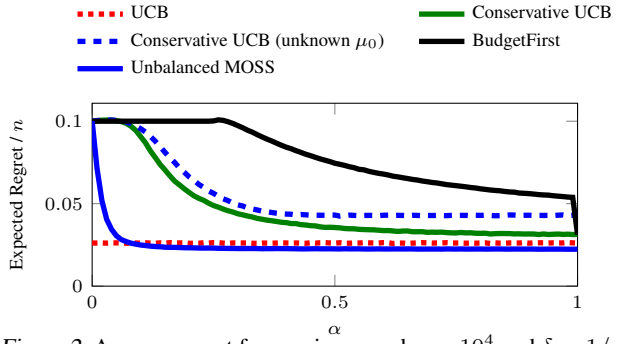

Figure 2. Average regret for varying α and $n = 10^4$ and $\delta = 1/n$

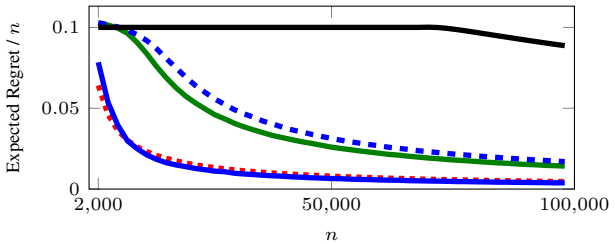

Figure 3. Average regret as n varies with $\alpha = 0.1$ and $\delta = 1/n$

7. Conclusion

We introduced a new family of multi-armed bandit frameworks motivated by the requirement of exploring conservatively to maintain revenue and demonstrated various strategies that act effectively under such constraints. We expect that similar strategies generalize to other settings, like contextual bandits and reinforcement learning. We want to emphasize that this is just the beginning of a line of research that has many potential applications. We hope that others will join us in improving the current results, closing open problems, and generalizing the model so it is more widely applicable.

Acknowledgments

This work was supported by Alberta Innovates – Technology Futures and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

R. Agrawal. Sample mean based index policies with $o(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995.

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

E. Even-Dar, M. Kearns, Y. Mansour, and J. Wortman. Regret to the best vs. regret to the average. *Machine Learning*, 72(1-2):21–37, 2008.

T. Gabel and M. Riedmiller. Distributed policy search reinforcement learning for job-shop scheduling tasks. *International Journal of Production Research*, 50(1):41–61, 2011.

J. García and F. Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

A. Garivier. Informational confidence bounds for self-normalized averages and applications. *arXiv preprint arXiv:1309.3376*, 2013.

M. Hutter and J. Poland. Adaptive online prediction by following the perturbed leader. *Journal of Machine Learning Research*, 6:639–660, 2005.

K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil'UCB: An optimal exploration algorithm for multi-armed bandits. In *COLT-2014*, pages 423—439, 2014.

M. N. Katehakis and H. Robbins. Sequential choice from several populations. *Proceedings of the National Academy of Sciences of the United States of America*, 92 (19):8584, 1995.

E. Kaufmann, A. Garivier, and O. Cappé. On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 2015. To appear.

W. M. Koolen. The Pareto regret frontier. In *Advances in Neural Information Processing Systems*, pages 863–871, 2013.

T. Lattimore. The Pareto regret frontier for bandits. In *Advances in Neural Information Processing Systems*, 2015a. To appear.

T. Lattimore. Optimally confident UCB : Improved regret for finite-armed bandits. Technical report, 2015b. URL http://arxiv.org/abs/1507.07880.

Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popović. Towards automatic experimentation of educational knowledge. In *SIGCHI Conference on Human Factors in Computing Systems (CHI 2014)*, pages 3349–3358. ACM Press, 2014.

G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3150–3158, 2015.

V. Rieser and O. Lemon. Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation. In *ACL-08: HLT*, pages 638–646, 2008.

A. Sani, G. Neu, and A. Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, pages 810–818, 2014.

Y. Sui, A. Gotovos, J. Burdick, and A. Krause. Safe exploration for optimization with Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 997–1005, 2015.