Regulating Greed Over Time

STEFANO TRACÀ  AND  CYNTHIA RUDIN

Abstract

In retail, there are predictable yet dramatic time-dependent patterns in customer behavior, such as periodic changes in the number of visitors, or increases in visitors just before major holidays (e.g., Christmas). The current paradigm of multi-armed bandit analysis does not take these known patterns into account, which means that despite the firm theoretical foundation of these methods, they are fundamentally flawed when it comes to real applications. This work provides a remedy that takes the time-dependent patterns into account, and we show how this remedy is implemented in the UCB and $\varepsilon$-greedy methods. In the corrected methods, exploitation (greed) is regulated over time, so that more exploitation occurs during higher reward periods, and more exploration occurs in periods of low reward. In order to understand why regret is reduced with the corrected methods, we present a set of bounds that provide insight into why we would want to exploit during periods of high reward, and discuss the impact on regret. Our proposed methods have excellent performance in experiments, and were inspired by a high-scoring entry in the Exploration and Exploitation 3 contest using data from Yahoo! Front Page. That entry heavily used time-series methods to regulate greed over time, which was substantially more effective than other contextual bandit methods. Keywords: Multi-armed bandit, exploration-exploitation trade-off, time series, retail management, marketing, online applications, regret bounds.

1 Introduction

Consider the classic pricing problem faced by retailers, where the price of a product is chosen to maximize the expected profit. The optimal price is learned through a mix of exploring various pricing choices and exploiting those known to yield higher profits. Exploration-exploitation problems occur not only in retail (both in stores and online), but in marketing, on websites such as Yahoo! Front Page, where the goal is to choose which of a set of articles to display to the user, on other webpages where ads are shown to the user on a sidebar (e.g., Facebook, Slashdot), and even on websites like YouTube that recommend the next video (and relative targeted ad) to watch. In all of these applications, the classic assumptions of random rewards with a static probability distribution is badly violated. For retailers, there are almost always clear trends in customer arrivals, and they are often predictable. For instance, Figure 1a shows clear weekly periodicity of customer arrivals, with a large dip on Christmas Eve and a huge peak on Boxing Day and Figure 1b shows clear half-day periodicity of the number of Skype users. These dramatic trends might have a substantial impact on which policy we would use to show ads and price products – if we knew how to handle them – yet these trends are completely ignored in classic multi-armed bandit analysis, where we assume each arm has a static fixed distribution of rewards.

The time series patterns within the application areas discussed above take many different forms, but in all cases, the same effect is present, where regulating the rate of exploitation (i.e., greed) over time could be beneficial. For retailers, the number of customers is much larger on certain days than others, and for these days we should exploit by choosing the best prices, and not explore. For Yahoo! Front Page, articles have a short lifespan and some articles are much better than others, in which case, if we find a particularly good article, we should exploit by repeatedly showing that one, and not explore new articles. For online advertising, there are certain periods where customers are more likely to make a purchase, so prices should be set to optimal values during those times, and we should not be exploring suboptimal choices.

In this work, we consider a simple but effective way to model a time-dependent effect on the rewards, which is that the static rewards for all the possible choices (the so-called arms, e.g. the ads shown on a website) are multiplied by a known time series, the reward multiplier $G(t)$. Even in this simple case, we can intuitively see how exploiting when the reward multiplier is high and exploring when it is low can lead to better performance. The reward multiplier approach is general, and can encode “micro-lock” (i.e., playing many times the same arm in one round) or “lock” periods (i.e., playing the same arm in a sequence of rounds) where the prices can not change, but the purchase rate has dynamically changing trends. As a result of the reward multiplier function, theoretical regret bound analysis of the multi-armed bandit problem becomes more complicated, because now the distribution of rewards depends explicitly on time. We not only care how many times each suboptimal arm is played, but exactly when they are played. For instance, if suboptimal arms are played only when the reward multiplier is low, intuitively it should not hurt the overall regret; this is one insight that the classic analysis cannot provide.
The standard algorithms are not able to be used for the setting in which the rewards are altered by the multiplier function (i.e. in case of “micro-lock” times). This is because they would incorrectly estimate the mean rewards of the arms.

The main contributions of this work are:

- A new framework that illustrates when it is beneficial to stop exploration sometimes to favor exploitation;
- Novel algorithms that show how to adapt existing policies to regulate greed over time:
  - Algorithm 1: $\varepsilon$-greedy algorithm with regulating threshold (Section 3.2);
  - Algorithm 2: soft $\varepsilon$-greedy algorithm (Section 3.3);
  - Algorithm 3: UCB algorithm with regulating threshold (Section 3.4);
  - Algorithm 4: soft UCB algorithm (Section 3.5).
- Theoretical regret bounds for the above algorithms;
- Numerical comparisons (in Section 4) with improved versions of the classic $\varepsilon$-greedy algorithm (Algorithm 6) and UCB algorithm (Algorithm 5).

The new algorithms, that take advantage of the shape of the multiplier function $G(t)$, are not a simple extension of the $\varepsilon$-greedy algorithm and the UCB algorithm. The new algorithms perform substantially better than even “smarter” versions of the $\varepsilon$-greedy algorithm and the UCB algorithm (see Algorithm 6 and Algorithm 5 in Section 4) where the rewards obtained at each round are properly calculated. This is because the algorithms we propose regulate greed over time.

The multiplier function is assumed to be known and bounded. If the function $G(t)$ is not known in advance, it may be easy to predict. For example, it may be easy to estimate the interest in a specific product at a particular day and time. Figure 1 shows a very clear trend for Google searches for “strawberries”.

---

(a) English users shopping online. Source: ispreview.co.uk

(b) Users on Skype. Source: voiceontheweb.biz
Moreover, it would be also possible to estimate $G(t)$ at each step $t$ of the algorithm, since the knowledge of the function in future steps is not necessary. Figure 2 shows the trend of Google searches for “scarf” in the United States. The higher demand for scarves in 2014 was predictable in the short term due to the particularly cold winter of that year.

The ideas in this paper were inspired by a high scoring entry in the Exploration and Exploitation 3 Phase 1 data mining competition, where the goal was to build a better recommendation system for Yahoo! Front Page news articles. At each time, several articles were available to choose from, and these articles would appear only for short time periods and would never be available again. One of the main ideas in this entry was simple yet effective: if any article gets more than 9 clicks out of the last 100 times we show the article, and keep displaying it until the clickthrough rate goes down. This alone increased the clickthrough rate by almost a quarter of a percent. The way we distilled the problem within this paper allows us to isolate and study this effect.

2 Related Work

The setup of this work differs from other works considering time-dependent multi-armed bandit problems – we do not assume the mean rewards of the arms exhibit random changes over time, and we assume that the reward multiplier is known in advance, in accordance with what we observe in real scenarios. No previous works that we know of consider regulating greed over time based on known reward trends. In contrast, Liu et al. [2013] consider a problem where each arm transitions in an unknown Markovian way to a different reward state when it is played, and evolves according to an unknown random process when it is not played. Garivier and Moulines [2008] presented an analysis of a discounted version of the UCB and a sliding window version of the UCB, where the distribution of rewards can have abrupt changes and stays stationary in between. Besbes et al. [2014] considers the case where the mean rewards for each arm can change, where the variation of that change is bounded. Slivkins and Upfal [2007] consider an extreme case where the rewards exhibit Brownian motion, leading to regret bounds that scale differently than typical bounds (linear in $T$ rather than logarithmic). One of the works that is extremely relevant to ours is that of Chakrabarti et al. [2009] who consider “mortal bandits” that disappear, just like Yahoo! Front Page news articles or online advertisements. The setting of mortal bandits in the Yahoo! dataset inspired the framework proposed here, since in the mortal setting, we claim one would want to curb exploration when a high reward mortal bandit is available. An interesting setting is discussed by Komiyama et al. [2013], where there are lock-up periods when one is required to play the same arm several times in a row. In our scenario, the micro-lock-up periods occur at each step of the game, and their effective lengths are given by $G(t)$. In some of the algorithms that we propose, when the rewards multiplier are above a certain threshold, we create lock-up periods ourselves, where exploration is stopped, but the best arm is allowed to change.
during high-reward zones as we gather information over rounds. The methods proposed in this work are contextual [Li et al. [2010]], in the sense that we consider externally available information in the form of time dependent trends.

3 Algorithms for regulating greed over time

This section illustrates the problem, the proposed algorithms to regulate greed over time, and theoretical results on the bound on the expected regret of each policy.

3.1 Problem setup

Formally, the stochastic multi-armed bandit problem with regulated greed is a game played in \( n \) rounds. At each round \( t \) the player chooses an action among a finite set of \( m \) possible choices called arms (for example, they could be ads shown on a website, recommended videos and articles, or prices). When arm \( j \) is played \( (j \in \{1, \cdots, m\}) \) an unscaled random reward \( X_j(t) \) is drawn from an unknown distribution and the player receives the scaled reward \( X_j(t)G(t) \) where \( G(t) \) is the multiplier function. The distribution of \( X_j(t) \) does not change with time (the index \( t \) is just used to indicate in which turn the reward was drawn), while \( G(t) \) is a known function of time assumed to be bounded (this is, for instance, the number of searches for a particular item on Google or the number of users on Skype). At each turn, the player suffers also a possible regret from not having played the best arm: the mean regret for having played arm \( j \) is given by \( \Delta_j = \mu_* - \mu_j \), where \( \mu_* \) is the mean reward of the best arm (indicated by “*”) and \( \mu_j \) is the mean reward obtained when playing arm \( j \). At the end of each turn the player can update her estimate of the mean reward of arm \( j \):

\[
\hat{X}_j = \frac{1}{T_j(t-1)} \sum_{s=1}^{T_j(t-1)} X_j(s),
\]

where \( T_j(t-1) \) is the number of times arm \( j \) has been played before round \( t \) starts. This update will hopefully help the player in choosing a good arm in the next round. The total regret at the end of the game is given by

\[
R_n = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{I_t=j\}},
\]

where \( \mathbb{1}_{\{I_t=j\}} \) is an indicator function equal to 1 if arm \( j \) is played at time \( t \) (otherwise its value is 0). The strategies presented in the following sections aim to minimize the expected cumulative regret \( \mathbb{E}[R_n] \) by regulating exploitation (i.e., greed) of the best arm found so far, and exploration based on the values of the multiplier function \( G(t) \). In general, when the multiplier function is high, the player risks to incur in high regret if a bad arm is played. We show that it is beneficial to stop exploration in this situation and resume exploration when rewards and regrets are lower. A complete list of the symbols used throughout the paper can be found in Appendix D.

3.2 Regulating greed with threshold in the \( \varepsilon \)-greedy algorithm

In Algorithm 1 we present a variation of the \( \varepsilon \)-greedy algorithm of [Auer et al. [2002]], in which a threshold \( z \) has been introduced in order to regulate greed. An optimal threshold \( z \) can also be estimated by running the algorithm on past data and by evaluating the one that gives the lowest regret. At each turn, \( t \), when the rewards are “high” (i.e., the \( G(t) \) multiplier is above the threshold \( z \)) the algorithm exploits the best arm found so far, that is, arm \( j \) with the highest mean estimate given in equation (1). When the rewards are “low” (i.e., the \( G(t) \) multiplier is under the threshold \( z \)), the algorithm will explore with probability \( \varepsilon_t = \min\{1, \frac{m}{T} \} \) an arm at random (each arm has probability \( 1/m \) of being selected). The number \( t \) counts how many times the multiplier function has been under the threshold up to time \( t \), while the constant \( k \) is greater than 10 and such that \( k \geq \frac{1}{\min_j \Delta_j} \). The reason of this choice is clear by looking at the expression of \( \beta_j(t) \) which is a bound on the probability of considering incorrectly a suboptimal arm \( j \) being the best choice. By setting the parameter \( k \) accordingly, we can ensure the logarithmic bound on the expected cumulative regret over the number of rounds (because the \( \varepsilon_t \) are \( \theta(1/t) \) and their sum over time is logarithmically bounded, while the \( \beta_j(t) \) term is \( o(1/t) \)).

The following theorem provides a bound on the mean regret of this policy (the proof is given in Appendix A).
Algorithm 1: $\varepsilon$-greedy algorithm with regulating threshold

\textbf{Input}: number of rounds $n$, number of arms $m$, threshold $\varepsilon$, a constant $k > 10$, such that $k > \frac{4}{\min_j \Delta_j}$.

\textbf{Initialization}: play all arms once and initialize $\hat{X}_j$ (defined in (1)) for each $j = 1, \ldots, m$ for $t = m + 1$ to $n$ do

if $(G(t) < z)$ then

with probability $\varepsilon_t$ play an arm uniformly at random (each arm has probability $\frac{1}{m}$ of being selected), otherwise (with probability $1 - \varepsilon_t$) play arm $j$ such that

$$\hat{X}_j \geq \hat{X}_i \forall i$$

else

play arm $j$ such that

$$\hat{X}_j \geq \hat{X}_i \forall i$$

end

Get reward $G(t)X_j$; Update $\hat{X}_j$; end

Theorem 3.1 ($\varepsilon$-greedy algorithm with hard threshold). The bound on the mean regret $\mathbb{E}[R_n]$ at time $n$ is given by

\[
\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j + \sum_{t=m+1}^{n} G(t) \mathbb{I}_{\{G(t) < z\}} \sum_{j: \mu_j < \mu_*} \Delta_j \left( \varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \beta_j(\hat{t}) \right) + \sum_{t=m+1}^{n} G(t) \mathbb{I}_{\{G(t) \geq z\}} \sum_{j: \mu_j < \mu_*} \Delta_j \beta_j(\hat{t}),
\]

where

\[
\beta_j(\hat{t}) = k \left( \frac{\hat{t}}{mke} \right)^{-\frac{n}{m}} \log \left( \frac{\hat{t}}{mke} \right) + \frac{4}{\Delta_j^2} \left( \frac{\hat{t}}{mke} \right)^{-\frac{n \Delta_j^2}{m}}.
\]

The sum in (3) is the exact mean regret during the initialization phase of Algorithm 1. In (4) we have a bound on the expected regret for turns that present low values of $G(t)$, where the quantity in the parenthesis is the bound on the probability of playing arm $j$: $\beta_j(\hat{t})$ is the bound on the probability that arm $j$ is considered being the best arm at round $t$, and $1/m$ is the probability of choosing arm $j$ when the choice is made at random. Finally, in (5) we have a bound on the expected regret for turns that present high values of $G(t)$ and in this case we consider only the probability $\beta_j(\hat{t})$ that arm $j$ is the best arm since we do not explore at random during high reward periods. The usual $\varepsilon$-greedy algorithm is a special case when $G(t) = 1 \forall t$ and $\varepsilon > 1$. Notice that $\varepsilon_t$ is a quantity $\theta(1/\hat{t})$, while $\beta_j(\hat{t})$ is $\Theta(1/\hat{t})$, so that an asymptotic logarithmic bound in $n$ holds for $\mathbb{E}[R_n]$ if $\hat{t}$ grows at the same rate as $t$ (because of the logarithmic bound on the harmonic series).

We want to compare this bound with the one of the usual version of the $\varepsilon$-greedy algorithm but, since the old version is not well suited for the setting in which the rewards are altered by the multiplier function, we discount the rewards obtained at each round (by simply dividing them by $G(t)$) so that it can also produce accurate estimates of the mean reward for each arm. This “smarter” version of the $\varepsilon$-greedy algorithm is presented in Algorithm 5 (Section 4). The bound on the probability of playing a suboptimal arm $j$ for the usual $\varepsilon$-greedy algorithm is given by $\beta_j(\hat{t})$ (i.e. $\beta_j(\hat{t})$ when $\hat{t} = t$) and we refer to it as $\beta_j^\text{old}(t)$. In general, $\beta_j^\text{old}(t)$ is lower than $\beta_j(\hat{t})$ (since $\hat{t} \leq t$). Intuitively, this reflects the fact that the new algorithm performs fewer exploration steps. Moreover, in the usual $\varepsilon$-greedy algorithm, the probability of choosing arm $j$ at time $t$ is given by

\[
P\left(\{I_t^\text{old} = j\}\right) = \varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \beta_j^\text{old}(t),
\]
which is less than the probability of the new algorithm in case of low $G(t)$

$$
P \left( \{ I_t^{\text{new}} = j \} \right) = \varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \beta_j(\tilde{t}),
$$

but can easily be higher than the probability of the new algorithm in case of high rewards (which is given by only $\beta_j(\tilde{t})$). In fact,

$$
P \left( \{ I_t^{\text{old}} = j \} \right) - P \left( \{ I_t^{\text{new}} = j \} \right) = \varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \beta_j(\tilde{t}) - \beta_j(\tilde{t}) = \frac{1}{m} \min \left\{ 1, \frac{km}{t} \right\} \beta_j(\tilde{t}) \min \left\{ 1, \frac{km}{t} \right\} + \beta_j(\tilde{t}) - \beta_j(\tilde{t}),
$$

if $t > km$ we get

$$
\frac{1}{m} \frac{km}{t} + \beta_j(\tilde{t}) \left( 1 - \frac{km}{t} \right) - \beta_j(\tilde{t}), \quad (7)
$$

if $t \leq km$ we get

$$
\frac{1}{m} - \beta_j(\tilde{t}) + \beta_j(\tilde{t}) - \beta_j(\tilde{t}), \quad (8)
$$

and for $t$ large enough both expressions are positive since $\beta_j(\tilde{t})$ is $o \left( \frac{1}{t} \right)$ and we assume that $\tilde{t}$ is $\theta(t)$. Having $[7]$ and $[8]$ positive means that if we are in a high-rewards period the probability of choosing a suboptimal arm decreases faster than the $\varepsilon$-greedy algorithm. In practice, the threshold $\varepsilon$ should be defined as $\arg\min(\mathbb{E}[R_n])$. If this is too computationally challenging, but past data are available, a good value for $\varepsilon$ can be chosen using cross validation techniques, i.e. by trying different thresholds with the available data and by choosing the one that yields the best performance.

The following Corollary illustrates the benefits of the bound in a simple scenario when the multiplier function can only take two values and the regulating threshold divides the higher value from the lower one.

**Corollary 3.1.** Suppose the greed function $G(t)$ takes only two values: $g_{\text{low}}$ and $g_{\text{high}}$. At each turn $t$ it takes the value $g_{\text{low}}$ for a fraction $q$ of the turns played, and the value $g_{\text{high}}$ for the remaining $t - qt$ turns (for example, if $q = 1/2$, $G(t)$ alternates at each turn between $g_{\text{low}}$ and $g_{\text{high}}$). Then, the bound on the expected regret at turn $n$ reduces to:

$$
\mathbb{E}[R_n] \leq O(1)
$$

$$
= \frac{k}{q} \Delta_{g_{\text{low}}} g_{\text{low}} \sum_{t=m+1}^{n} \mathbb{1}\{G(t) = g_{\text{low}}\} \left( \frac{1}{t} + o \left( \frac{1}{t} \right) \right) + \Delta_{g_{\text{high}}} g_{\text{high}} \sum_{t=m+1}^{n} \mathbb{1}\{G(t) = g_{\text{high}}\} \mathcal{O} \left( \frac{1}{t} \right),
$$

where $\Delta_{\text{tot}} = \sum_{j : \mu_j < \mu_*} \Delta_j$.

The term that hurts regret the most $(1/t)$ is multiplied only by $g_{\text{low}}$, and not by $g_{\text{high}}$. When the rewards are high (and so is the possible regret), only terms of order $o(1/t)$ are present. If exploration were permitted during the high reward zone, there would have been large terms of $g_{\text{high}}/t$, which is what the algorithm is designed to avoid.

### 3.3 Soft $\varepsilon$-greedy algorithm

We present in Algorithm 2 a “soft version” of the $\varepsilon$-greedy algorithm where greed is regulated gradually (in contrast with the hard threshold of the previous section). Again, in high reward zones, exploitation will be preferred, while in low reward zones the algorithm will explore the arms more. Let us define the following function

$$
\psi(t) = \frac{\log \left( 1 + \frac{1}{G(t)} \right)}{\log \left( 1 + \frac{1}{\min_{s \in \{m+1, \ldots, n\}} G(s)} \right)},
$$

and let $\gamma = \min_{s \in \{m+1, \ldots, n\}} \psi(s)$. Notice that $0 < \psi(t) \leq 1 \forall t$ and that its values are close to 0 when $G(t)$ is high, while they are close to 1 for low values of $G(t)$. The new probabilities of exploration during the game are given at each turn $t$ by
\[ \varepsilon_t = \min \{ \psi(t), \frac{km}{t} \} \]. In this way, we still maintain the linear decay of the probabilities of exploration, but we push them to zero to avoid high regrets when the multiplier function \( G(t) \) is high. We generally assume that \( \min_{s \in \{m+1, \ldots, n\}} G(s) \) is not smaller than 1. The usual case is recovered when \( G(t) = 1 \) for all \( t \).

**Algorithm 2:** Soft \( \varepsilon \)-greedy algorithm

| Input | number of rounds \( n \), number of arms \( m \), a constant \( k > 10 \), such that \( k > \frac{4}{\min_j \Delta_j} \), sequences \( \{\varepsilon_t\}_{t=1}^n \) = \{ \psi(t), \frac{km}{t} \} \) and \( \{G(t)\}_{t=1}^n \) |
|-------|---------------------------------------------------------------|
| **Initialization:** | play all arms once and initialize \( \hat{X}_j \) (defined in (1)) for each \( j = 1, \ldots, m \) |
| for \( t = m + 1 \) to \( n \) do | With probability \( \varepsilon_t \) play an arm uniformly at random (each arm has probability \( \frac{1}{m} \) of being selected), otherwise (with probability \( 1 - \varepsilon_t \)) play arm \( j \) such that \( \hat{X}_j \geq \hat{X}_i \) \( \forall i \) |
| Get reward \( G(t)X_j \) |
| Update \( \hat{X}_j \) |
| end |

The following theorem (proved in Appendix [A]) shows that a logarithmic bound holds in this case too (because the \( \varepsilon_t \) are \( \theta(1/t) \) and their sum over time is logarithmically bounded, while the \( \beta_j^S(t) \) term is \( o(1/t) \)).

**Theorem 3.2** (Regret-bound for soft-\( \varepsilon \)-greedy algorithm). The bound on the mean regret \( \mathbb{E}[R_n] \) at time \( n \) is given by

\[
\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j + \sum_{t=m+1}^{n} G(t) \sum_{j:\mu_j < \mu_*} \Delta_j \left( \varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \beta_j^S(t) \right) \quad (11)
\]

where

\[
\beta_j^S(t) = k \left( \frac{\gamma t}{mke} \right)^{-\frac{m}{k}} \log \left( \frac{\gamma t}{mke} \right) + \frac{4}{\Delta_j^2} \left( \frac{\gamma t}{mke} \right)^{-\frac{4m^2}{k^2}}. \quad (12)
\]

The sum in (10) is the exact mean regret during the initialization of Algorithm 2. For the rounds after the initialization phase, the quantity in the parenthesis of (11) is the bound on the probability of playing arm \( j \) (where \( \beta_j^S(t) \) is the bound on the probability that arm \( j \) is the best arm at round \( t \), and \( 1/m \) is the probability of choosing arm \( j \) when the choice is made at random).

As before, we want to compare this bound with the “smarter” version of the \( \varepsilon \)-greedy algorithm presented in Algorithm 6. In the usual \( \varepsilon \)-greedy algorithm, after the “critical time” \( n' = km \), the probability \( \mathbb{P}(\hat{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)}) \) of arm \( j \) being the current best arm, can be bounded by a quantity \( \beta_j^{old}(t) \) that is \( o(1/t) \) as \( t \) grows. Before time \( n' \), the decay of \( \mathbb{P}(\hat{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)}) \) is faster and the bound is a quantity that is \( o(1/t^\lambda) \), \( \forall \lambda \) as \( t \) grows (see Remark 1 in Appendix A). The probability of choosing a suboptimal arm \( j \) changes as follows:

- if \( t < n' \), \( \mathbb{P}(\{I_t = j\}) = \frac{1}{m} \);  
- if \( t \geq n' \), \( \mathbb{P}(\{I_t = j\}) = \frac{k}{t} + (1 - \frac{km}{t}) \beta_j^{old}(t) \), which is \( \theta(\frac{1}{t}) \) as \( t \) grows.

In the soft-\( \varepsilon \)-greedy algorithm, before time \( w \) defined as \( w = \arg\min f(s) \), subject to \( f(s) < \gamma \), where \( f(s) = \frac{km}{s} \), we have that \( \beta_j^S(t) \), which is the bound on the probability \( \mathbb{P}(\hat{X}_{j,T_j(t)} \geq \hat{X}_{i,T_i(t)}) \) of arm \( j \) being the current best arm, is a quantity that is \( o(1/(\gamma t)^\lambda) \), \( \forall \lambda \) as \( t \) grows (the argument is similar to the Remark 1 in Appendix A). After \( w \), it can be bounded by a quantity that is \( o(1/(\gamma t)) \) as \( t \) grows. The probability of choosing a suboptimal arm \( j \) changes as follows:

- if \( t < n' \), \( \mathbb{P}(\{I_t = j\}) = \frac{1}{m} \psi(t) + (1 - \psi(t)) \beta_j^S(t) \);  
- if \( n' \leq t \leq w \), \( \mathbb{P}(\{I_t = j\}) = \frac{1}{m} \min \{ \psi(t), \frac{km}{t} \} + (1 - \min \{ \psi(t), \frac{km}{t} \}) \beta_j^S(t) \);  
- if \( t > w \), \( \mathbb{P}(\{I_t = j\}) = \frac{1}{m} \min \{ \psi(t), \frac{km}{t} \} \beta_j^S(t) \).
• if \( t > w \), \( \mathbb{P}(\{I_t = j\}) = \frac{k}{t} + \left(1 - \frac{k}{t}\right) \beta^S_j(t) \).

In order to interpret these quantities, let us see what happens for high or low values of the multiplier \( G(t) \) as \( t \) grows in Table 1. For brevity, we abuse notation when using Landau’s symbols, because in some cases \( t \) is not allowed to go to infinity; it is convenient to still use the “little o” notation to compare the decay rates of the probabilities of choosing a suboptimal arm, which also gives a qualitative explanation of what happens when using the algorithms. For the soft-\( \varepsilon \)-algorithm, the rate at which the probability of choosing a suboptimal arm decays is faster when \( G(t) \) is high, and worse when \( G(t) \) is low. Notice that the parameter \( \gamma \) slows down the decay with respect to the usual \( \varepsilon \)-greedy algorithm. This is a direct consequence of the slower exploration. An example of a typical behavior of \( \psi(t) \) and \( \varepsilon_t^{old} \) is shown in Figure 3, where \( G(t) = 20 + 19 \sin(t/2) \).

Table 1: Summary of the decay rate of the probabilities of choosing a suboptimal arm for the soft-\( \varepsilon \)-greedy algorithm and the usual \( \varepsilon \)-greedy algorithm (supposing it is taking in account the time-patterns.) The decay depends on the time-regions of the game presented in Figure 3.

| Region | round | \( G(t) \) | \( \mathbb{P}(\{I_t = j\})^{old} \) | \( \mathbb{P}(\{I_t = j\})^{soft} \) | \( \mathbb{P}(\{I_t = j\})^{soft} < \mathbb{P}(\{I_t = j\})^{old} \) ? |
|--------|-------|-------------|-------------------------------|-------------------------------|------------------------------------------------------|
| 1      | high  | \( \frac{1}{m} \) | \( o\left(\frac{1}{(\gamma t)^\lambda}\right), \forall \lambda \) | yes, much better |
|        | low   | \( \frac{1}{m} \) | close to \( \frac{1}{m} \) | no, but not by much |
| 2      | high  | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{t}\right) \) | \( o\left(\frac{1}{(\gamma t)^\lambda}\right), \forall \lambda \) | yes, much better |
|        | low   | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{t}\right) \) | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{(\gamma t)^\lambda}\right), \forall \lambda \) | yes, but not by much |
| 3      | high  | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{t}\right) \) | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{(\gamma t)^\lambda}\right) \) | no, but not by much |
|        | low   | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{t}\right) \) | \( \theta \left(\frac{1}{t}\right) + o\left(\frac{1}{(\gamma t)^\lambda}\right) \) | no, but not by much |

Figure 3: Comparison of probabilities of exploration over the number of rounds. Before \( n' \), \( \varepsilon_t^{old} \) is 1 and always greater than \( \psi(t) \). After \( w \), \( \varepsilon_t^{old} \) is always less than \( \psi(t) \).
3.4 Regulating greed in the UCB algorithm

Following what has been presented to improve the ε-greedy algorithm in this setting, we introduce in Algorithm 3 a modification of the UCB algorithm. We again set a threshold \( z \) and, if the multiplier of the rewards \( G(t) \) is above this level, the new algorithm exploits the best arm. When \( G(t) \) is under the threshold, the algorithm is going to play the arm with the highest upper confidence bound on the mean estimate.

\[ \text{Algorithm 3: UCB algorithm with regulating threshold} \]

\begin{verbatim}
Input: number of rounds \( n \), number of arms \( m \), threshold \( z \), sequence \( \{G(t)\}_{t=1}^{n} \)
Initialization: play all arms once and initialize \( \hat{X}_{j} \) (as defined in (1)) for each \( j = 1, \cdots, m \)
for \( t = m + 1 \) to \( n \) do
    if \( (G(t) < z) \) then
        play arm \( j \) with the highest upper confidence bound on the mean estimate
        \[ \hat{X}_{j, \tau_{j}(t-1)} + \sqrt{\frac{2 \log t}{T_{j}(t-1)}}; \]
    else
        play arm \( j \) such that
        \[ \hat{X}_{j} \geq \hat{X}_{i} \forall i; \]
    end
end
Get reward \( G(t)X_{j}; \)
Update \( \hat{X}_{j}; \)
\end{verbatim}

It is possible to prove that also in this case the regret can be bounded logarithmically in \( n \). Let \( B = \{ t : G(t - 1) < z, G(t) > z \} \) be the set of rounds where the high-reward zone is entered, and let \( \tau_{1} \) be the last round of the high-reward zone that was entered at time \( t \). Let us call \( y_{1}, y_{2}, \cdots, y_{B} \) the elements of \( B \) and order them in increasing order such that \( y_{1} < y_{2} < \cdots < y_{B} \). Let us also define for every \( k \in \{1, \cdots, |B|\} \) the set \( Y_{k} = \{ t : t \geq y_{k}, G(t) > z, t < y_{k+1} \} \) (where \( y_{B+1} = n \)) of times in the high-reward period entered at time \( y_{k} \), and let \( \Lambda_{k} = \max_{t \in Y_{k}} G(t) \) the highest value of \( G(t) \) on \( Y_{k} \). Finally, for every \( k \), let \( R_{k} = \Lambda_{k}|Y_{k}| \).

Now, given a game of \( n \) total rounds, we can “collapse” the \( k \)th high reward zone into the entering time \( y_{k} \) by defining \( G(y_{k}) = R_{k} \), for all \( k \). Now, the maximum regret over \( B \) is given by \( (\max_{j} \Delta_{j}) \sum_{k=1}^{|B|} R_{k} \). By eliminating the set \( B \) from the game, we have transformed the original game into a shorter one, with \( \eta \) steps, where \( G(t) \) is bounded by \( z \) and the usual UCB algorithm is played. When the size of set \( B \) decreases with \( n \), (is of order \( \theta(1/t) \) after an arbitrary time), the total regret has a logarithmic bound in \( n \).

The ε-greedy methods are more amenable to this type of analysis than UCB methods, because the proofs require bounds on the probability of choosing the wrong arm at each turn. The UCB proof instead require us to bound the expected number of times the suboptimal arms are played, without regard to when those arms were chosen. We were able to avoid using the maximum of the \( G(t) \) values in the ε-greedy proofs, but this is unavoidable in the UCB proofs without leaving terms in the bound that cannot be explicitly calculated or simplified (an alternate proof would use weaker Central Limit Theorem arguments).
Theorem 3.3 (Regret-bound for the regulated UCB algorithm). The bound on the mean regret $\mathbb{E}[R_n]$ at time $n$ is given by

$$\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j \quad (13)$$

$$+ z \left[ 8 \sum_{j: \mu_j < \mu_*} \frac{\log \eta}{\Delta_j} + \left(1 + \frac{\pi^2}{3}\right) \left(\sum_{j=1}^{m} \Delta_j\right) \right] \quad (14)$$

$$+ \left(\max_j \Delta_j\right) \sum_{k=1}^{|B|} r_k \quad (15)$$

The first sum in (13) is the exact mean regret of the initialization phase of Algorithm 3, the third sum in (15) is the bound on the regret from the high-reward zones that have been collapsed, and the second term in (14) is the bound on the regret for $\eta$ rounds when $G(t)$ is under the threshold $z$ and it follows from the usual bound on the UCB algorithm (for $n$ rounds the UCB algorithm has a mean regret bounded by $\sum_{j=1}^{m} \log \frac{n}{\Delta_j} + \left(1 + \frac{\pi^2}{3}\right) \sum_{j=1}^{m} \Delta_j$).

Again, the threshold $z$ should be defined as $\text{argmin}(\mathbb{E}[R_n])$ or, if past data are available, $z$ can be chosen using cross validation.

3.5 The soft UCB algorithm

In Algorithm 4, present now a “soft version” of the UCB algorithm where greed is regulated gradually (in contrast with the hard threshold of the previous section). Again, in high reward zones, exploitation will be preferred, while in low reward zones the algorithm will explore the arms. Let us define the following function:

$$\xi(t) = \left(1 + \frac{t}{G(t)}\right) \quad (16)$$

At each turn $t$ of the game, the algorithm plays the arm with the highest upper confidence bound on the mean estimate, but, with the introduction of $\xi(t)$, the confidence interval around $\hat{X}_{j,T_j(t-1)}$ is built in a way such that, when $G(t)$ is high, it collapses on the estimate itself, forcing the player to choose the arm with the highest mean estimate (thus, leading to a pure exploitation policy). In contrast, when the multiplier $G(t)$ is low, the confidence interval around $\hat{X}_{j,T_j(t-1)}$ stretches out, making the player explore more easily arms with high uncertainty.

One of the main difficulties of the formulation of these bounds is to define a correct functional form for $\xi(t)$ so that it is possible to obtain smoothness in the arm decision, reasonable Chernoff-Hoeffding inequality bounds while working out the proof (see Appendix C), and a convergent series (the second summation in (18)).

| Algorithm 4: Soft UCB algorithm |
|---------------------------------|
| **Input**: number of rounds $n$, number of arms $m$, sequence $\{G(t)\}_{t=1}^{n}$ |
| **Initialization**: play all arms once and initialize $\hat{X}_j$ (as defined in (11)) for each $j = 1, \cdots, m$ |
| for $t = m + 1$ to $n$ do |
| play arm $j$ with the highest upper confidence bound on the mean estimate: $\hat{X}_{j,T_j(t-1)} + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}}$ |
| Get reward $G(t)X_j$; |
| Update $\hat{X}_j$; |
| end |

Also in this case, it is possible to achieve a bound that grows logarithmically in $n$.  

10
Theorem 3.4 (Regret-bound for soft-UCB algorithm). The bound on the mean regret $\mathbb{E}[R_n]$ at time $n$ is given by

$$
\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j + \max_{t \in \{m+1, \ldots, n\}} G(t) \left( \sum_{j, \mu_j < \mu_*} \frac{8}{\Delta_j} \log \left( \frac{\max_{t \in \{m+1, \ldots, n\}} \xi(t)}{\Delta_j} \right) + \sum_{j=1}^{m} \Delta_j \left[ 1 + \frac{n}{\sum_{t=m+1}^{n} 2\xi(t)^{-4}(t-1-m)^2} \right] \right). \tag{18}
$$

The first sum in (17) is the exact mean regret of the initialization phase of Algorithm 4. For the rounds after the initialization phase, the mean regret is bounded by the quantity in (18), which is almost identical to the bound of the usual UCB algorithm if we assume $G(t) = 1$ (i.e., rewards are not modified by the multiplier function).

4 Experimental results

We consider three types of multiplier function $G(t)$:

- The Wave Greed (Figure 4a): in this case customers come in waves: $G(t) = 21 + 20 \sin(0.25t)$. We want to exploit the best arm found so far during the peaks, and explore the other arms during low-rewards periods;

- The Christmas Greed (Figure 4b): again, $G(t) = 21 + 20 \sin(0.25t)$, but when $t \in \{650, 651, \ldots, 670\}$, $G(t) = 1000$ which shows that there is a peak in the rewards offered by the game (which we call “Christmas”, in analogy to the phenomenon of the boom of customers during the Christmas holidays);

- The Step Greed (Figure 4c): this case is similar to the Wave Greed case, but this time the function is not smooth: $G(t) = 200$, but for $t \in \{600, 601, \ldots, 800\} \cup \{1000, 1001, \ldots, 1200\} \cup \{1400, 1402, \ldots, 1600\}$ we have $G(t) = 400$.

Figure 4: Shapes of the multiplier functions used in the experiments.

(a) The Wave Greed  
(b) The Christmas Greed  
(c) The Step Greed

We consider a game with 500 arms and normally distributed rewards. Each arm $j \in \{1, \ldots, 500\}$ has mean reward $\mu_j = 0.1 + (200 + 1.5(500 - j + 1))/(1.5 \times 500)$ and common standard deviation $\sigma = 0.05$. The arms were chosen in this way so that $X_j$ would take values (with high probability) in $[0, 1]$. Having a bounded support for $X_j$ is a standard assumption made when proving regret bounds (see Auer et al. [2002]). We play 2000 rounds each game. After 2000 rounds the algorithms all essentially have determined which arm is the best and tend to perform very similarly from that point onwards.

The well-known UCB and $\epsilon$-greedy algorithms are not suitable for the setting in which the rewards are altered by the multiplier function. Thus, in their current form, we can not compare directly with them. The fact that rewards are multiplied would irremediably bias all the estimations of the mean rewards, leading UCB and $\epsilon$-greedy to choose arms that look good just because they happened to be played in a high reward period. For example, suppose we show an ad on a website at lunch time: many people will see it because at that time the web-surfing is at its peak (i.e., the $G(t)$ multiplier is high). So even if the ad was bad, we may register more clicks than a good ad showed at 3:00AM (i.e., the $G(t)$ multiplier is low). To obtain a fair comparison, we created “smarter” versions of the UCB and $\epsilon$-greedy algorithms in which the rewards are discounted at...
each round (by simply dividing them by $G(t)$) so that also the old version of the algorithms can be smarter in that they can produce accurate estimates of the mean reward for each arm. The smarter version of the usual UCB algorithm is presented in Algorithm 5 and the one for the $\varepsilon$-greedy algorithm is shown in Algorithm 6. For the three multiplier functions, we report the performance of the algorithms in Figures 5, 6, and 7.

In Figures 8, 9, and 10, we change the rewards to have a Bernoulli distribution (the assumption of bounded support is verified). Similarly to the normal case, each arm $j \in \{1, \ldots, 500\}$ has probability of success $p_j = 0.1 + (200 + 1.5(500 - j + 1))/(1.5 \times 500)$. One of the advantages of the $\varepsilon$-greedy algorithm is that there are no assumptions on the distribution of the rewards, while in UCB they need bounded support ([0, 1] for convenience, so it is easier to use Hoeffding’s inequality).

**Figure 5:** Comparison for the **Wave Greed** case.

(a) $\varepsilon$-greedy algorithms rewards.

(b) UCB algorithms rewards.

(c) Final rewards comparison

**Figure 6:** Comparison for the **Christmas Greed** case.

(a) $\varepsilon$-greedy algorithms rewards.

(b) UCB algorithms rewards.

(c) Final rewards comparison

**Figure 7:** Comparison for the **Step Greed** case.

(a) $\varepsilon$-greedy algorithms rewards.

(b) UCB algorithms rewards.

(c) Final rewards comparison
**Algorithm 5:** Smarter version of the usual UCB algorithm

| Input | number of rounds $n$, number of arms $m$, sequence $\{G(t)\}_{t=1}^n$ |
|-------|------------------------------------------------------------------|
| Initialization | play all arms once and initialize $\hat{X}_j$ (as defined in (1)) for each $j = 1, \ldots, m$ |
| for $t = m + 1$ to $n$ do |
| \hspace{1cm} play arm $j$ with the highest upper confidence bound on the mean estimate: |
| \hspace{2cm} $\hat{X}_{j,T_j(t-1)} + \sqrt{\frac{2 \log(t)}{T_j(t-1)}}$ |
| Get reward $G(t)X_j$; |
| Update $\hat{X}_j$; |
| end |

**Algorithm 6:** Smarter version of the usual $\varepsilon$-greedy algorithm

| Input | number of rounds $n$, number of arms $m$, a constant $c > 10$, a constant $d$ such that $d < \min_j \Delta_j$ and $0 < d < 1$, sequences $\{\varepsilon_t\}_{t=1}^n = \min \{1, \frac{\log n}{m}\}$ and $\{G(t)\}_{t=1}^n$ |
|-------|------------------------------------------------------------------|
| Initialization | play all arms once and initialize $\hat{X}_j$ (as defined in (1)) for each $j = 1, \ldots, m$ |
| for $t = m + 1$ to $n$ do |
| \hspace{1cm} with probability $\varepsilon_t$ play an arm uniformly at random (each arm has probability $\frac{1}{m}$ of being selected), |
| \hspace{2cm} otherwise (with probability $1 - \varepsilon_t$) play arm $j$ such that |
| \hspace{3cm} $\hat{X}_j \geq \hat{X}_i \ \forall i$ |
| Get reward $G(t)X_j$; |
| Update $\hat{X}_j$; |
| end |
4.1 Discussion on Yahoo! contest

The motivation of this work comes from a high scoring entry in the Exploration and Exploitation 3 contest, where the goal was to build a better recommender system for Yahoo! Front Page news article recommendations. The contest data, which was from Yahoo! and allows for unbiased evaluations, is described by [Li et al., 2010]. These data had several challenging characteristics, including broad trends over time in click through rate, arms (news articles) appearing and disappearing over time, the inability to access the data in order to cross-validate, and other complexities. This paper does not aim to handle all of these, but only the one which led to a key insight in increased performance, which is the regulation of greed over time. Although there were features available for each time, none of the contestants were able to successfully use the features to substantially boost performance, and the exploration/exploitation aspects turned out to be more important. Here are the main
insights leading to large performance gains, all involving regulating greed over time:

- “Peak grabber”: Stop exploration when a good arm appears. Specifically, when the article was clicked 9/100 times, keep showing it and stop exploration all together until the arm’s click through rate drops below that of another arm. Since this strategy does not handle the massive global trends we observed in the data, it needed to be modified as follows:

- “Dynamic peak grabber”: Stop exploration when the click through rate of one arm is at least 15% above that of the global click through rate.

- Stop exploring old articles: We can determine approximately how long the arm is likely to stay, and we reduce exploration gradually as the arm gets older.

- Do not fully explore new arms: When a new arm appears, do not use 1 as the upper confidence bound for the probability of click, which would force a UCB algorithm to explore it, use .88 instead. This allows the algorithm to continue exploiting the arms that are known to be good rather than exploring new ones.

The peak grabber strategies inspired the abstracted setting here, where one can think of a good article appearing during periods of high $G(t)$, where we would want to limit exploration; however, the other strategies are also relevant cases where the exploration/exploitation tradeoff is regulated over time. There were no “lock-up” periods in the contest dataset, though as discussed earlier, the $G(t)$ function is also relevant for modeling that setting. The large global trends we observed in the contest data click through rates are very relevant to the $G(t)$ model, since obviously one would want to explore less when the click rate is high in order to get more clicks overall.

5 Conclusions

The dynamic trends we observe in most retail and marketing settings are dramatic. It is possible that understanding these dynamics and how to take advantage of them is central to the success of multi-armed bandit algorithms. We showed in this work how to adapt regret bound analysis to this setting, where we now need to consider not only how many times an arm was pulled in the past, but precisely when the arm was pulled. The key element of our algorithms is that they regulate greed (exploitation) over time, where during high reward periods, less exploration is performed.

There are many possible extensions to this work. In particular, if $G(t)$ is not known in advance, it may be easy to estimate from data in real time, as in the dynamic peak grabber strategy. The analysis of the algorithms in this paper could be extended to other important multi-armed bandit algorithms besides $\epsilon$-greedy and UCB. Further, future work will consider the connection of mortal bandits (with appearing/disappearing arms) with the $G(t)$ setting, since for mortal bandits, each bandit’s $G(t)$ function can change at a different rate.

Acknowledgments

We thank team members Ed Su and Virot Ta Chiraphadhanakul for their work on the ICML Exploration and Exploitation 3 contest that inspired the work here. Thank you also to Philippe Rigollet for helpful comments and encouragement. This project was partially funded by the Ford-MIT alliance.
References

Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. *arXiv preprint arXiv:1302.6927*, 2013.

Jean-Yves Audibert, Sébastien Bubeck, et al. Best arm identification in multi-armed bandits. *Conference on Learning Theory 2010-Proceedings*, 2010.

Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica, 61*(1-2):55–65, 2010.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning, 47*(2-3):235–256, 2002.

Donald A Berry and Bert Fristedt. *Bandit Problems: Sequential Allocation of Experiments (Monographs on Statistics and Applied Probability)*. Springer, 1985.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. *Available at SSRN 2436629*, 2014.

Olivier Bousquet. New approaches to statistical learning theory. *Annals of the Institute of Statistical Mathematics, 55*(2):371–389, 2003. ISSN 0020-3157. doi: 10.1007/BF02530506. URL [http://dx.doi.org/10.1007/BF02530506](http://dx.doi.org/10.1007/BF02530506).

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 273–280, 2009.

Aurélien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. *arXiv preprint arXiv:1102.2490*, 2011.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600, 2012.

Junpei Komiyama, Issei Sato, and Hiroshi Nakagawa. Multi-armed bandit problem with lock-up periods. In *Asian Conference on Machine Learning*, pages 116–132, 2013.

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*, pages 661–670. ACM, 2010.

Haoyang Liu, Keqin Liu, and Qing Zhao. Learning in a changing world: Restless multiarmed bandit with unknown dynamics. *IEEE Transactions on Information Theory, 59*(3):1902–1916, 2013.

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

Cynthia Rudin, Virot Ta Chiraphadhanakul, and Edward Su. Regulating greed over time for Yahoo! front page news article recommendations, from the Exploration and Exploitation 3 Challenge. lecture slides, 2012.

Aleksandrs Slivkins and Eli Upfal. Adapting to a stochastically changing environment: The dynamic multi-armed bandits problem. Technical Report CS-07-05, Brown University, 2007.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.
A Regret-bound for $\varepsilon$-greedy algorithm with hard threshold

The regret at round $n$ is given by

$$R_n = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{I_t = j\}},$$

(19)

where $G(t)$ is the greed function evaluated at time $t$, $\mathbb{1}_{\{I_t = j\}}$ is an indicator function equal to 1 if arm $j$ is played at time $t$ (otherwise its value is 0) and $\Delta_j = \mu^* - \mu_j$ is the difference between the mean of the best arm reward distribution and the mean of the $j$'s arm reward distribution. By considering the threshold $z$ which determines which rule is applied to decide what arm to play, we can rewrite the regret as

$$R_n = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{G(t) < z\}} \mathbb{1}_{\{I_t = j\}} + \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{G(t) \geq z\}} \mathbb{1}_{\{I_t = j\}}.$$

By taking the expectation we have that

$$\mathbb{E}[R_n] = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{G(t) < z\}} \mathbb{P}(\{I_t = j\}) + \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{G(t) \geq z\}} \mathbb{P}(\{I_t = j\}),$$

which can be rewritten as

$$\mathbb{E}[R_n] = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{G(t) < z\}} \left[ \frac{\varepsilon_t}{m} + (1 - \varepsilon_t) \mathbb{P}(\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{i,T_i(t-1)} \forall i) \right] + \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{G(t) \geq z\}} \mathbb{P}(\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{i,T_i(t-1)} \forall i).$$

(20)

For the rounds of the algorithm where $G(t) < z$, we are in the standard setting, so for those times, we follow the standard proof of Auer et al. (2002). For the times that are over the threshold, we need to create a separate bound. Let us now bound the probability of playing the sub-optimal arm $j$ at time $t$ when the greed function is above the threshold $z$.

$$\mathbb{P}(\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{i,T_i(t-1)} \forall i) \leq \mathbb{P}(\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{*_{T_j(t-1)}}) \leq \mathbb{P}(\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}) + \mathbb{P}(\tilde{X}_{*_{T_j(t-1)}} \leq \mu_* - \frac{\Delta_j}{2}).$$

(21)

(22)

where the last inequality follows from the fact that

$$\{\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{*_{T_j(t-1)}}\} \subset \left(\{\tilde{X}_{*_{T_j(t-1)}} \leq \mu_* - \frac{\Delta_j}{2}\} \cup \{\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\}\right).$$

(23)

In fact, suppose that there exist an element $\omega \in \{\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{*_{T_j(t-1)}}\}$ that does not belong to

$$\left(\{\tilde{X}_{*_{T_j(t-1)}} \leq \mu_* - \frac{\Delta_j}{2}\} \cup \{\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\}\right).$$

Then, we would have that

$$\omega \in \left(\{\tilde{X}_{*_{T_j(t-1)}} \leq \mu_* - \frac{\Delta_j}{2}\} \cup \{\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\}\right)^c \left(\{\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\} \cap \{\tilde{X}_{j,T_j(t-1)} < \mu_j + \frac{\Delta_j}{2}\}\right).$$

(24)

(25)

but from the intersection of events given in (25) it follows that $\tilde{X}_{*_{T_j(t-1)}} \geq \mu_* - \frac{\Delta_j}{2} = \mu_j + \frac{\Delta_j}{2} > \tilde{X}_{j,T_j(t-1)}$ which contradicts $\omega \in \{\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{*_{T_j(t-1)}}\}$. 

17
Therefore, all elements of $\{\tilde{X}_{j,T_j(t-1)} \geq \tilde{X}_{s,T_s(t-1)}\}$ belong to $\left(\{\tilde{X}_{s,T_s(t-1)} \leq \mu_s - \frac{\Delta_s}{2}\} \cup \{\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\}\right)$. Let us consider the first term of (22) (the computations for the second term are similar),

$$
\mathbb{P}\left(\tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right) = \sum_{s=1}^{t-1} \mathbb{P}\left(T_j(t-1) = s, \tilde{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right)
= \sum_{s=1}^{t-1} \mathbb{P}\left(T_j(t-1) = s|\tilde{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) \mathbb{P}\left(\tilde{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right)
\leq \sum_{s=1}^{t-1} \mathbb{P}\left(T_j(t-1) = s|\tilde{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) e^{-\frac{\Delta_j^2}{2} s},
$$

(26)

where in the last inequality we used the Chernoff-Hoeffdings bound. Let us define $T_j^R(t-1)$ as the number of times arm $j$ is played at random (note that $T_j^R(t-1) \leq T_j(t-1)$ and that $T_j^R(t-1) = \sum_{s=1}^{t-1} B_s$ where $B_s$ is a Bernoulli r.v. with parameter $\varepsilon_s/m$), and let us define

$$
x_0 = \frac{1}{2m} \sum_{s=1}^{\tilde{t}} \varepsilon_s,
$$

where $\tilde{t}$ is the number of rounds played under the threshold $z$ up to time $t$. Then,

$$
\sum_{s=1}^{\lfloor x_0 \rfloor} \frac{\varepsilon_s}{m} \leq \frac{1}{2m} \sum_{s=1}^{\tilde{t}} \varepsilon_s
$$

(26)

$$
\sum_{s=1}^{\lfloor x_0 \rfloor} \mathbb{P}\left(T_j^R(t-1) \leq \lfloor x_0 \rfloor\right) = \sum_{s=1}^{\tilde{t}} \mathbb{P}\left(T_j^R(t-1) \leq \lfloor x_0 \rfloor\right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} \lfloor x_0 \rfloor}
\leq \lfloor x_0 \rfloor \mathbb{P}\left(T_j^R(t-1) \leq \lfloor x_0 \rfloor\right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} \lfloor x_0 \rfloor}
$$

(27)

where for the first $\lfloor x_0 \rfloor$ terms of the sum we upperbounded $e^{-\frac{\Delta_j^2}{2} s}$ by 1, and for the remaining terms we used the fact that $\sum_{\lfloor x_0 \rfloor+1}^{\infty} e^{-ks} \leq \frac{1}{k} e^{-kx}$, where in our case $k = \frac{\Delta_j^2}{2}$. We have that

$$
\mathbb{E}[T_j^R(t-1)] = \frac{1}{m} \sum_{s=1}^{\tilde{t}} \varepsilon_s, \quad Var(T_j^R(t-1)) = \sum_{s=1}^{\tilde{t}} \frac{\varepsilon_s}{m} \left(1 - \frac{\varepsilon_s}{m}\right) \leq \frac{1}{m} \sum_{s=1}^{\tilde{t}} \varepsilon_s = \mathbb{E}[T_j^R(t-1)],
$$

and, using the Bernstein inequality $\mathbb{P}(S_n \leq \mathbb{E}[S_n] - a) \leq \exp\{-\frac{a^2/2}{\mathbb{E}[S_n]+a/2}\}$ with $S_n = T_j^R(t-1)$ and $a = \frac{1}{2} \mathbb{E}[T_j^R(t-1)]$,

$$
\mathbb{P}(T_j^R(t-1) \leq \lfloor x_0 \rfloor) = \mathbb{P}\left(T_j^R(t-1) \leq \mathbb{E}[T_j^R(t-1)] - \frac{1}{2} \mathbb{E}[T_j^R(t-1)]\right)
\leq \exp\left\{-\frac{1}{8} (\mathbb{E}[T_j^R(t-1)])^2\right\}
= \exp\left\{-\frac{1}{8} \mathbb{E}[T_j^R(t-1)]\right\} = \exp\left\{-\frac{1}{8} \mathbb{E}[T_j^R(t-1)]\right\} = \exp\left\{-\frac{1}{8} \mathbb{E}[T_j^R(t-1)]\right\}.
$$

(28)
Now we need a lower bound on \( |x_0| \). Let us define \( n' = \lfloor km \rfloor \), then

\[
x_0 = \frac{1}{2m} \sum_{s=1}^{m} \varepsilon_s
\]

\[
= \frac{1}{2m} \sum_{s=1}^{n'} \min \left\{ 1, \frac{km}{s} \right\}
\]

\[
= \frac{1}{2m} \sum_{s=1}^{n'} 1 + \frac{1}{2m} \sum_{s=n'+1}^{n'} \frac{km}{s}
\]

\[
= \frac{n'}{2m} + \frac{1}{2m} \left( \sum_{s=1}^{n'} \frac{km}{s} - \sum_{s=1}^{n'} \frac{km}{s} \right)
\]

\[
\geq \frac{n'}{2m} + \frac{k}{2} \left( \log(\tilde{t} + 1) - (\log(n') + \log(e)) \right)
\]

\[
\geq \frac{k}{2} \log \left( \frac{n'}{m} \right) + \frac{k}{2} \log \left( \frac{i}{n'e} \right)
\]

\[
= \frac{k}{2} \log \left( \frac{i}{mke} \right). \tag{29}
\]

**Remark 1.** Note that if \( \tilde{t} \) (or \( t \) in the usual \( \varepsilon \)-greedy algorithm) was less than \( n' \), then we would have \( x_0 = \frac{\tilde{t}}{2m} \), yielding an exponential decay of the bound on the probability of \( j \) being the best arm. To see this, \( \tilde{t} < n' \) would imply that, using (27) and (28),

\[
\leq \frac{\tilde{t}}{2m} \exp \left\{ -\frac{1}{2} \frac{\tilde{t}}{m} \right\} + \frac{2}{\Delta_j^2} \exp \left\{ -\Delta_j^2 \frac{\tilde{t}}{2m} \right\}
\]

Continuing the proof of Theorem 3.1, we obtain a bound on the first term in (22) as follows. Using (29) combined with (28) in (27), we get that

\[
\frac{k}{2} \left( \frac{\tilde{t}}{mke} \right)^{-\frac{1}{\Delta_j}} \log \left( \frac{\tilde{t}}{mke} \right) + \frac{4}{\Delta_j^2} \left( \frac{\tilde{t}}{mke} \right)^{-\frac{k+2}{4}}. \tag{30}
\]

Since the computations for the second term in (22) are similar, a bound on \( \Pr(\hat{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)} \forall i) \) is given by

\[
\beta_j(\tilde{t}) = k \left( \frac{\tilde{t}}{mke} \right)^{-\frac{1}{\Delta_j}} \log \left( \frac{\tilde{t}}{mke} \right) + \frac{4}{\Delta_j^2} \left( \frac{\tilde{t}}{mke} \right)^{-\frac{k+2}{4}}. \tag{31}
\]

We have now an upper bound for \( \Pr(\hat{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)} \forall i) \). We can use this to easily bound \( \Pr(\hat{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)} \forall i) \) in (20) which yields the following bound on the mean regret at time \( n \):

\[
\mathbb{E}[R_n] \leq \sum_{j=1}^{n} G(j) \Delta_j
\]

\[
+ \sum_{t=m+1}^{n} G(t) \mathbb{I}_{\{G(t) < z\}} \sum_{j: \mu_j \leq \mu_*} \Delta_j \left( \frac{\varepsilon_t}{m} + (1 - \varepsilon_t) \beta_j(\tilde{t}) \right)
\]

\[
+ \sum_{t=m+1}^{n} G(t) \mathbb{I}_{\{G(t) \geq z\}} \sum_{j: \mu_j \leq \mu_*} \Delta_j \beta_j(\tilde{t}),
\]

This, combined with the bound \( \beta_j(\tilde{t}) \) above, proves the theorem.

**B Logarithmic bound for Soft-\( \varepsilon \)-greedy algorithm**

At each round \( t \), arm \( j \) is played with probability \( \frac{\varepsilon_t}{m} + (1 - \varepsilon_t) \Pr(\hat{X}_j \geq \hat{X}_i \forall i) \)
where $\varepsilon_t = \min \left\{ \psi(t), \frac{km}{t} \right\}$ and
\[
\psi(t) = \frac{\log \left( 1 + \frac{1}{G(t)} \right)}{\log \left( 1 + \frac{1}{\min_{s \in \{m+1, \ldots, n\}} G(s)} \right)}.
\] (32)

Recall that $\gamma = \min_{1 \leq t \leq n} \psi(t)$. Let us bound the probability $P(\{I_t = j\})$ of playing the sub-optimal arm $j$ at time $t$. We have that
\[
P \left( \tilde{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)} \ \forall i \right) \leq P \left( \tilde{X}_{j,T_j(t-1)} \geq \hat{X}_{s,T_s(t-1)} \right)
\]
\[
\leq P \left( \tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2} \right) + P \left( \tilde{X}_{j,T_j(t-1)} \leq \mu_s + \frac{\Delta_j}{2} \right).
\] (33)

For the bound on the two addends in (33), we have identical steps to the proof for Theorem 1, and thus
\[
P \left( \tilde{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2} \right) \leq |x_0|P \left( T_j^R(t-1) \leq |x_0| \right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} |x_0|}.
\] (34)
\[
P \left( \tilde{X}_{j,T_j(t-1)} \leq \mu_s + \frac{\Delta_j}{2} \right) \leq |x_0|P \left( T_j^R(t-1) \leq |x_0| \right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} |x_0|}.
\] (35)

and we again have
\[
P (T_j^R(t-1) \leq |x_0|) \leq \exp \left\{ -\frac{1}{5} |x_0| \right\}.
\] (36)

Now we need a lower bound on $|x_0|$. Let $t > w$ where $w = \min\{1, \ldots, n\}$ such that $\frac{\gamma m}{\sqrt{w}} < \gamma$. Then,
\[
x_0 = \frac{1}{2m} \sum_{s=1}^{t} \varepsilon_s
\]
\[
= \frac{1}{2m} \sum_{s=1}^{t} \min \left\{ \psi(s), \frac{km}{s} \right\}
\]
\[
\geq \frac{1}{2m} \sum_{s=1}^{w} \gamma + \frac{1}{2m} \left( \sum_{s=1}^{t} \frac{km}{s} - \sum_{s=1}^{w} \frac{km}{s} \right)
\]
\[
\geq \frac{w \gamma}{2m} + \frac{k}{2} \left( \log(t+1) - \log(w) + \log(e) \right)
\]
\[
\geq \frac{k}{2} \log \left( \frac{w \gamma}{m \ k} \right) + \log \left( \frac{t}{we} \right)
\]
\[
= \frac{k}{2} \log \left( \frac{\gamma t}{mke} \right).
\] (37)

Using (37) in (36), combined with (34) and (35), from (33) the bound on $P (\tilde{X}_{j,T_j(t-1)} \geq \hat{X}_{i,T_i(t-1)} \ \forall i)$ is given by
\[
\beta_j^S(t) = k \log \left( \frac{\gamma t}{mke} \right) \left( \frac{\gamma t}{mke} \right)^{-\frac{k}{m}} + \frac{4}{\Delta_j^2} \left( \frac{\gamma t}{mke} \right)^{-\frac{\Delta_j^2}{4}}.
\]

Since the mean regret is given by
\[
\mathbb{E}[R_n] = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j G(t) P (\{I_t = j\}),
\] (38)
the bound on the mean regret at time $n$ is given by
\[
\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j
\]
\[
+ \sum_{t=m+1}^{n} G(t) \sum_{j=1}^{m} \Delta_j \left( \varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \beta_j^S(t) \right).
\]
C The regret bound of the soft UCB algorithm

The regret at round $n$ is given by

$$R_n = \sum_{j=1}^{m} G(j) \Delta_j + \sum_{t=m+1}^{n} \sum_{j=1}^{m} \Delta_j G(t) \mathbb{1}_{\{I_t=j\}} \leq \sum_{j=1}^{m} G(j) \Delta_j + \left( \max_{t \in \{m+1, \ldots, n\}} G(t) \right) \sum_{j=1}^{m} \sum_{t=m+1}^{n} \Delta_j \mathbb{1}_{\{I_t=j\}}.$$

The expected regret $\mathbb{E}[R_n]$ at round $n$ is bounded by

$$\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j + \left( \max_{t \in \{m+1, \ldots, n\}} G(t) \right) \sum_{j=1}^{m} \Delta_j \mathbb{E}[T_j(n)].$$

where $T_j(n) = \sum_{t=1}^{n} \mathbb{1}_{\{I_t=j\}}$ is the number of times the sub-optimal arm $j$ has been chosen up to round $n$. Recall from (1) that

$$\hat{X}_j = \frac{1}{T_j(t-1)} \sum_{s=1}^{T_j(t-1)} X_j(s).$$

From the Chernoff-Hoeffding Inequality we have that

$$P \left( \frac{1}{T_j(t-1)} \sum_{i=1}^{T_j(t-1)} X_{j,i} - \mu_j \leq -\varepsilon \right) \leq \exp\{-2T_j(t-1)\varepsilon^2\},$$

and

$$P \left( \frac{1}{T_j(t-1)} \sum_{i=1}^{T_j(t-1)} X_{j,i} - \mu_j \geq \varepsilon \right) \leq \exp\{-2T_j(t-1)\varepsilon^2\}. \quad (41)$$

Let us define the following function:

$$\xi(t) = \left( 1 + \frac{t}{G(t)} \right),$$

by selecting $\varepsilon = \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}}$ we have

$$P \left( \hat{X}_j + \frac{2 \log \xi(t)}{T_j(t-1)} \leq \mu_j \right) \leq \xi(t)^{-4}, \quad (42)$$

and

$$P \left( \hat{X}_j - \frac{2 \log \xi(t)}{T_j(t-1)} \geq \mu_j \right) \leq \xi(t)^{-4}. \quad (43)$$

Equivalently, we may write for every $j$

$$\mu_j - \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \leq \hat{X}_j \quad \text{with probability at least} \quad 1 - \xi(t)^{-4},$$

$$\mu_j + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \geq \hat{X}_j \quad \text{with probability at least} \quad 1 - \xi(t)^{-4}. \quad (44)$$

If we choose arm $j$ at round $t$ (i.e., the event $\{I_t=j\}$ occurs) we have that

$$\hat{X}_j + \frac{2 \log \xi(t)}{T_j(t-1)} \geq \hat{X}_j + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \geq \hat{X}_j + \frac{2 \log \xi(t)}{T_j(t-1)}.$$
Let us use (45) to upper bound the LHS and (43) to lower bound the RHS of (46), then we get with probability at least \(1 - 2\xi(t)^{-4}\)

\[
\mu_j + 2\sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \geq \mu_*,
\]

from which we get with probability at least \(1 - 2\xi(t)^{-4}\)

\[
T_j(t-1) \leq \frac{8}{\Delta_j^2} \log \xi(t). \tag{47}
\]

In order to emphasize the dependence of \(\hat{X}_j\) from \(T_j(t-1)\) we will sometimes write \(\hat{X}_{j,T_j(t-1)}\). In the following, notice that in (49) the summation starts from \(m + 1\) because in the first \(m\) initialization rounds each arm is played once. Moreover, step (50) follows from (49) by assuming that arm \(j\) has already been played \(u\) times. By using (46) we get (51), then, for each \(t\),

\[
\left\{ \hat{X}_{j,T_j(t-1)} + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \geq \hat{X}_{*,T,(t-1)} + \sqrt{\frac{2 \log \xi(t)}{T_*(t-1)}}, \quad T_j(t-1) \geq u \right\} \subset \left\{ \max_{s_j \in \{u,\ldots,T_j(t-1)\}} \hat{X}_{j,s} + \frac{2 \log \xi(t)}{s_j} \geq \min_{s_* \in \{1,\ldots,T_*(t-1)\}} \hat{X}_{*,s_*} + \frac{2 \log \xi(t)}{s_*} \right\} \tag{48}
\]

which justifies (52). We also have that (48) is included in

\[
\bigcup_{s_* = 1}^{T_*(t-1)} \bigcup_{s_j = u}^{T_j(t-1)} \left\{ \hat{X}_{j,s} + \sqrt{\frac{2 \log \xi(t)}{s_j}} \geq \hat{X}_{*,s_*} + \sqrt{\frac{2 \log \xi(t)}{s_*}} \right\}. \tag{53}
\]

Thus, for any integer \(u\), we may write

\[
T_j(n) = 1 + \sum_{t = m + 1}^n 1\{I_t = j\} \tag{49}
\]

\[
u + \sum_{t = m + 1}^n 1\{I_t = j, T_j(t-1) \geq u\} \tag{50}
\]

\[
u + \sum_{t = m + 1}^n 1\left\{ \hat{X}_{j,T_j(t-1)} + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \geq \hat{X}_{*,T,(t-1)} + \sqrt{\frac{2 \log \xi(t)}{T_*(t-1)}}, \quad T_j(t-1) \geq u \right\} \tag{51}
\]

\[
u + \sum_{t = m + 1}^n 1\left\{ \max_{s_j \in \{u,\ldots,T_j(t-1)\}} \hat{X}_{j,s} + \frac{2 \log \xi(t)}{s_j} \geq \min_{s_* \in \{1,\ldots,T_*(t-1)\}} \hat{X}_{*,s_*} + \frac{2 \log \xi(t)}{s_*} \right\} \tag{52}
\]

\[
u + \sum_{t = m + 1}^n \sum_{s_* = 1}^{T_*(t-1)} \sum_{s_j = u}^{T_j(t-1)} 1\left\{ \hat{X}_{j,s} + \sqrt{\frac{2 \log \xi(t)}{s_j}} \geq \hat{X}_{*,s_*} + \sqrt{\frac{2 \log \xi(t)}{s_*}} \right\}. \tag{53}
\]

When

\[
1 \left\{ \hat{X}_j + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \geq \hat{X}_* + \sqrt{\frac{2 \log \xi(t)}{T_*(t-1)}} \right\} \tag{54}
\]

is equal to one, at least one of the following has to be true:

\[
\hat{X}_* \leq \mu_* - \sqrt{\frac{2 \log \xi(t)}{T_*(t-1)}}, \tag{55}
\]

\[
\hat{X}_j \geq \mu_j + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}}, \tag{56}
\]

\[
\mu_* \leq \mu_j + 2\sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}}. \tag{57}
\]

(In fact, suppose none of them hold simultaneously. Then from (55) we would have that \(\hat{X}_* > \mu_* - \sqrt{\frac{2 \log \xi(t)}{T_*(t-1)}}\); then, by applying (57) (with opposite verse since we are assuming it does not hold) we get \(\hat{X}_* > \mu_j + 2\sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} - \sqrt{\frac{2 \log \xi(t)}{T_*(t-1)}}\) and}
then from (56) (again, with opposite verse) follows that $\tilde{X}_s > \tilde{X}_j + \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} - \sqrt{\frac{2 \log \xi(t)}{T_s(t-1)}}$ which is in contradiction with (54). Now, if we set $u = \left[ \frac{8}{\Delta_j} \log \left( \max_{t \in \{m+1, \ldots, n\}} \xi(t) \right) \right]$, for $T_j(t-1) \geq u$,

$$
\mu_\ast - \mu_j - 2 \sqrt{\frac{2 \log \xi(t)}{T_j(t-1)}} \\
\geq \mu_\ast - \mu_j - 2 \sqrt{\frac{2 \log \xi(t)}{u}} \\
= \mu_\ast - \mu_j - \Delta_j \sqrt{\frac{\log \xi(t)}{\log \left( \max_{t \in \{m+1, \ldots, n\}} \xi(t) \right)}} \\
\geq \mu_\ast - \mu_j - \Delta_j = 0,
$$

therefore, with this choice of $u$, (57) can not hold.

Thus, using (55), we have that

$$
T_j(n) \leq \left[ \frac{8}{\Delta_j} \log \left( \max_{t \in \{m+1, \ldots, n\}} \xi(t) \right) \right] \\
+ \sum_{t=m+1}^{n} \sum_{s_s=1}^{T_s(t-1)} \sum_{s_j=u}^{T_j(t-1)} 1 \left\{ \tilde{X}_{s, s_s} \leq \mu_\ast - \sqrt{\frac{2 \log \xi(t)}{s_s}} \right\} \\
+ \sum_{t=m+1}^{n} \sum_{s_s=1}^{T_s(t-1)} \sum_{s_j=u}^{T_j(t-1)} 1 \left\{ \tilde{X}_{j, s_j} \geq \mu_j + \sqrt{\frac{2 \log \xi(t)}{s_j}} \right\}
$$

and by taking expectation,

$$
\mathbb{E}[T_j(n)] \leq \left[ \frac{8}{\Delta_j} \log \left( \max_{t \in \{m+1, \ldots, n\}} \xi(t) \right) \right] \\
+ \sum_{t=m+1}^{n} \sum_{s_s=1}^{T_s(t-1)} \sum_{s_j=u}^{T_j(t-1)} \mathbb{P} \left\{ \tilde{X}_{s, s_s} \leq \mu_\ast - \sqrt{\frac{2 \log \xi(t)}{s_s}} \right\} \\
+ \sum_{t=m+1}^{n} \sum_{s_s=1}^{T_s(t-1)} \sum_{s_j=u}^{T_j(t-1)} \mathbb{P} \left\{ \tilde{X}_{j, s_j} \geq \mu_j + \sqrt{\frac{2 \log \xi(t)}{s_j}} \right\} \\
\leq \frac{8}{\Delta_j} \log \left( \max_{t \in \{m+1, \ldots, n\}} \xi(t) \right) + 1 + 2 \sum_{t=m+1}^{n} \xi(t)^{-4}(t-1-m)^2.
$$

where in the last step we upperbound $T_s(t-1)$ and $T_j(t-1)$ by $(t - 1 - m)$ (cases where we have only played the best arm or arm $j$). Therefore, by using (39)

$$
\mathbb{E}[R_n] \leq \sum_{j=1}^{m} G(j) \Delta_j \\
+ \max_{t \in \{m+1, \ldots, n\}} G(t) \left( \sum_{j: \mu_j < \mu_\ast} \frac{8}{\Delta_j} \log \left( \max_{t \in \{m+1, \ldots, n\}} \xi(t) \right) + \sum_{j=1}^{m} \Delta_j \left[ 1 + \sum_{t=m+1}^{n} 2 \xi(t)^{-4}(t-1-m)^2 \right] \right).
$$
## D Notation summary

- $m$: number of arms;
- $n$: number of rounds;
- $G : \{1, \cdots, n\} \to \mathbb{R}^+$: known multiplier function;
- $X_j(t)$: unscaled random reward for playing arm $j$;
- $X_j(t)G(t)$: actual reward;
- $\mu^*$: mean reward of the optimal arm ($\mu^* = \max_{1 \leq j \leq m} \mu_j$);
- $\Delta_j$: difference between the mean reward of the optimal arm and the mean reward of arm $j$ ($\Delta_j = \mu^* - \mu_j$);
- $\hat{X}_j$: current estimate of $\mu_j$;
- $I_t$: arm played at turn $t$;
- $T_j(t-1)$: number of times arm $j$ has been played before round $t$ starts;
- $z$: threshold (used in Algorithm 1 and Algorithm 3);
- $\hat{\eta}_t$: number of rounds under the threshold $z$ up to time $t$;
- $k$: a constant greater than 10 such that $k > \frac{4}{\min_j \Delta_j}$ in Algorithm 1 and Algorithm 2;
- $c$: a constant greater than 10 in Algorithm 6;
- $d$: a constant such that $d < \min_j \Delta_j$ and $0 < d < 1$ in Algorithm 6;
- $\varepsilon_t$: probability of exploration at turn $t$ (used in Algorithm 1 and Algorithm 2);
- $\beta_j(t)$: upper bound on the probability of considering suboptimal arm $j$ being the best arm at round $t$ when using Algorithm 1;
- $\beta_{old_j}(t)$: upper bound on the probability of considering suboptimal arm $j$ being the best arm at round $t$ when using Algorithm 6;
- $\beta_{S_j}(t)$: upper bound on the probability of considering suboptimal arm $j$ being the best arm at round $t$ when using Algorithm 2;
- $\psi(t)$: smoothing function used to define the probabilities of exploration $\varepsilon_t$ in Algorithm 2 (see Figure 3);
- $\gamma$: lowest value of $\psi(t)$ ($\gamma = \min_{s \in \{m+1, \cdots, n\}} \psi(s)$);
- $n'$: particular time defined as $km$ in the comparison between Algorithm 2 and Algorithm 6 in Section 3.3;
- $w$: first round when $\frac{km}{w}$ is less than $\gamma$ ($w = \text{argmin} f(s)$, subject to $f(s) < \gamma$, where $f(s) = \frac{km}{s}$) in the comparison between Algorithm 2 and Algorithm 6 in Section 3.3;
- $B$: set of rounds when the “high reward” zone is entered in Algorithm 3 ($B = \{t : G(t-1) < z, G(t) > z\}$);
- $Y_k = \{t : t \geq y_k, G(t) > z, t < y_{k+1}\}$: set of rounds in the high-reward period entered at time $y_k$ ($k \in \{1, \cdots, |B|\}$) in Algorithm 3;
- $\Lambda_k = \max_{t \in Y_k} G(t)$: highest value of $G(t)$ on $Y_k$ in Algorithm 3;
- $R_k = \Lambda_k |Y_k|$: the maximum regret of the $k$th high reward zone in Algorithm 3;
- $\xi(t)$: smoothing function used to define the decision rule in Algorithm 4.
- $e$: Euler’s number;
- $R_n$: total regret at round $n$. 

---

24