Reducing exploration of dying arms in mortal bandits

STEFANO TRACÀ AND CYNTHIA RUDIN

Abstract

Mortal bandits have proven to be extremely useful for providing news article recommendations, running automated online advertising campaigns, and for other applications where the set of available options changes over time. Previous work on this problem showed how to regulate exploration of new arms when they have recently appeared, but they do not adapt when the arms are about to disappear. Since in most applications we can determine either exactly or approximately when arms will disappear, we can leverage this information to improve performance: we should not be exploring arms that are about to disappear. We provide adaptations of algorithms, regret bounds, and experiments for this study, showing a clear benefit from regulating greed for arms that will soon disappear. The dataset used to illustrate numerical performance on articles recommendation is the Yahoo! Front Page Today Module User Click Log Dataset.
Keywords: Multi-armed bandit, exploration-exploitation trade-off, retail management, online applications, regret bounds.

1 Introduction

In many applications of multi-armed bandits, the bandits are *mortal*, meaning that they do not exist for the full period over which the algorithm is running. In advertising, ads and coupons can come and go; in news article recommendation, the news is perpetually changing; in website optimization, the content changes to keep viewers interested. Chakrabarti et al. [2009] introduced and formalized the notion of mortal bandits, and there has been a body of work following this. This work has proved to be valuable in the setting of advertising (see Agarwal et al. [2009] and Feraud et al.) and in other areas such as communications underlaying cellular networks (see Maghsudi and Stańczak [2015]). In Bnaya et al. [2013] the authors propose an adaptation to the mortal settings of the popular UCB algorithm introduced by Auer et al. [2002]. While these algorithms are designed to adapt exploration based on when arms *appear*, they do not adapt when arms *disappear* (for example, in Bnaya et al. [2013] new arms are immediately played, even for arms that may soon die, which could be a poor strategy). In strategic implementations of mortal bandits, *we should not be exploring arms that are soon going to disappear*.

In the applications discussed above (advertising, news article recommendation, website optimization) and others, we know in advance when arms will appear or disappear. For coupons and discount sales, we launch them for known periods of time (e.g., a one day sale), whereas for news articles, we could choose to place them in a pool of possible featured articles for mobile devices for one day or one week. If the lifespans of the arms are not known, they can often be estimated. For instance, we can observe the distribution of the lifespans of the arms to determine when an arm is old relative to other arms. Alternatively, external features can be used to estimate the remaining lifespan of an arm. This work provides algorithms for the mortal bandit setting that reduce exploration for dying arms. In Section 2 we introduce two algorithms: the AG-L algorithm (adaptive greedy with life regulation) and the UCB-L algorithm (UCB mortal with life regulation). We present finite time regret bounds and in Section 3 we show experiments with a variety of arm distributions and properties. We also introduce the LinUCB-L algorithm (LinUCB mortal with life regulation) that can be used in a contextual setting or when arms present features.

2 Algorithms for regulating greed over arm life.

Formally, the mortal stochastic multi-armed bandit problem is a game played in n rounds. At each round t the algorithm chooses an action I_t among a finite set M_t of possible choices called *arms* (for example, they could be ads shown on a website, recommended videos and articles, or prices). When arm $j \in M_t$ is played, a random reward $X_j(t)$ is drawn from an unknown distribution. The distribution of $X_j(t)$ does not change with time (the index t is used to indicate in which turn the reward was drawn) and it is bounded in $[a, b]$ (and we denote with r the range $r = b - a$), while the set M_t can change: arms may become unavailable (they "die") or new arms may arrive (they "are born"). At each turn, the player suffers a possible regret from not having played the best arm: the mean regret for having played arm j at turn t is given by $\Delta_{j,i_t^*} = \mu_{i_t^*} - \mu_j$, where $\mu_{i_t^*}$ is the mean reward of the best arm available at turn t (indicated by i_t^*) and μ_j is the mean reward obtained when

playing arm j. Let us call $I(j)$ the set of turns during which the algorithm chose arm j. At the end of each turn the algorithm updates the estimate of the mean reward of arm j:

$$\widehat{X}_j = \frac{1}{T_j(t-1)} \sum_{s \in I(j)}^{T_j(t-1)} X_j(s), \tag{1}$$

where $T_j(t-1)$ is the number of times arm j has been played before round t starts.

Let us define M_t as the set of all available arms at turn t (M_1 is the starting set of arms). $M_I \subset M_1, M_I \neq \emptyset$ is the set of arms that are initialized (i.e., the algorithm plays all of them once before applying other policies.) The quantity that a policy tries to minimize is the cumulative regret R_n that is given by

$$R_n = \sum_{j \in M_I} \Delta_{j,i_t^*} + \sum_{t=m_I+1}^{n} \sum_{j \in M_t} \Delta_{j,i_t^*} \mathbb{1}_{\{t \in I(j)\}}, \tag{2}$$

where $\mathbb{1}_{\{t \in I(j)\}}$ is an indicator function equal to 1 if arm j is played at time t (otherwise its value is 0). The first summation in (2) is the regret that the algorithm suffers during the initialization phase when each arm in M_I is pulled once yielding a regret of Δ_{j,i_t^*}. For the rest of the game ($t \in \{m_I + 1, \cdots, n\}$), the algorithm incurs in Δ_{j,i_t^*} regret at time t only when arm j is available ($j \in M_t$) and it is pulled ($t \in I(j)$). Let us call $M = \bigcup_{t=1}^{n} M_t$ the set of all arms that appear during the game and $L_j = \{s_j, s_j + 1, \cdots, l_j\}$ the set of turns that arm j is available. Then, we can also write (2) as

$$R_n = \sum_{j \in M_I} \Delta_{j,i_t^*} + \sum_{j \in M} \sum_{\substack{t \in L_j \\ t > m_I}} \Delta_{j,i_t^*} \mathbb{1}_{\{t \in I(j)\}}. \tag{3}$$

Depending on the algorithm used, one formulation is more convenient than the other when computing a bound on the expected cumulative regret $\mathbb{E}[R_n]$. A complete list of the symbols used throughout the paper can be found in Appendix E.

2.1 The adaptive greedy with life regulation (AG-L) algorithm

In Algorithm 1 we introduce a modified version of the adaptive greedy algorithm (which we abbreviate with AG) presented in Chakrabarti et al. [2009]. We call this new version the adaptive greedy with life regulation algorithm, which we abbreviate with AG-L. AG-L handles rewards bounded in $[a, b]$, and regulates exploration based on the remaining life of the arms (that is, the algorithm avoids exploring arms that are going to disappear soon). During the initialization phase, the algorithm plays each arm in the initialization pool M_I once, where the initialization pool is chosen by the user. Then, to determine whether to explore arms, AG-L draws a Bernoulli r.v. with parameter

$$p = 1 - \frac{\max_{j \in M_t} \widehat{X}_j - a}{b - a},$$

which intuitively means that if the algorithm has a good available arm (i.e., an arm that has a high mean estimate) the probability of exploring a new arm is very low and the algorithm will play the best available arm so far (ignoring also arms that were excluded in the initialization phase or have been born and never played). If the value of the Bernoulli r.v. is 1, then AG-L proceeds by playing an arm at random among those arms whose remaining life is long enough: we call this set $M_t(\mathcal{L})$. One way to set $M_t(\mathcal{L})$ is to pick all arms in M_t such that their remaining lifespan is in the top 30% of the distribution of all remaining lifespans. $M_t(\mathcal{L})$ can also contain arms that have never been played before or that were excluded in the

initialization phase.

Algorithm 1: AG-L algorithm

Input : number of rounds n, initialization set of arms M_I, set M_t of available arms at time, rewards range $[a, b]$
Initialization: play all arms in M_I once, and initialize $\widehat{X}_j$ for each $j = 1, \cdots, m_I$
for $t = m_I + 1$ **to** n **do**
  Draw a Bernoulli B r.v. with parameter

$$p = 1 - \frac{\max_{j \in M_t} \widehat{X}_j - a}{b - a};$$

  if $B = 1$ **then**
    Play an arm at random from $M_t(\mathcal{L})$;
    else
      Play an arm j with highest $\widehat{X}_j$;
    end
  end
  Get reward $X_j(t)$;
  Update $\widehat{X}_j$;
end

In order to derive a finite time regret bound we introduce $\mathcal{H}_{t-1}$ as the set of all possible histories (after deterministic initialization) of the game up to turn $t-1$:

$$\mathcal{H}_{t-1} = \left\{ h = \begin{bmatrix} b_{m_I+1} & b_{m_I+2} & \dots & b_{t-1} \\ i_{m_I+1} & i_{m_I+2} & \dots & i_{t-1} \end{bmatrix} : b_s \in \{0,1\},\ i_s \in M_s, \quad \forall s \in \{m_I+1, \dots, t-1\} \right\}. \tag{4}$$

Each element h of $\mathcal{H}_{t-1}$ is a possible history of pulls before turn t and tells exactly what arm was pulled and if it was an exploration turn or an exploitation turn. If $b_s = 1$ we say that the algorithm explored at time s, if $b_s = 0$ we say that the algorithm exploited at time s, while i_s is the index of the arm that was played at time s. Let us define the linear transformation $g(p) = b + (a-b)p$ (used to standardize rewards in the interval $[0,1]$) and use a result from Vaughan and Venables [1972] for the PDF (or PMF) $f_{M(h,s)}(g(p))$ of the maximum of the estimated mean rewards at time s given that each arm has been pulled according to history h up to time $s-1$:

$$f_{M(h,k)}(x) = \frac{1}{(m_t - 1)!} \operatorname{perm} \left(\begin{bmatrix} F_1(x) & F_2(x) & \dots & F_{m_k}(x) \\ \vdots & \vdots & \vdots & \vdots \\ F_1(x) & F_2(x) & \dots & F_{m_k}(x) \\ f_1(x) & f_2(x) & \dots & f_{m_k}(x) \end{bmatrix} \right) \Bigg\} \ m_k - 1 \text{ rows},$$

where $f_1(x), \cdots, f_{m_k}(x)$ and $F_1(x), \cdots, F_{m_k}(x)$ are the PDFs (or PMFs) of the distributions of the average rewards (which we can compute knowing the distribution from which rewards are drawn). For each h, we indicate how many times arm j has been pulled up to time k with

$$t_j(h,k) = \mathbb{1}_{\{j \in M_I\}} + \sum_{s=m_I+1}^{k} \mathbb{1}_{\{i_s \in I(j)\}}.$$

Similarly to when we defined the regret, let us call $\Delta(i, i_s) = \mu_i - \mu_{i_s}$. Then, consider the following quantities (see Appendix A for how to compute them given the mean rewards):

- $u_s(h, i_s)$ is an upper bound on the probability that arm i_s is considered to be the best arm at time s given the history of pulls (according to h) up to time $s-1$:

$$u_s(h, i_s) = \prod_{i:\mu_i > \mu_{i_s}} \left(\exp\left\{ -\frac{t_{i_s}(h,s)\Delta(i,i_s)^2}{2r} \right\} + \exp\left\{ -\frac{t_i(h,s)\Delta(i,i_s)^2}{2r} \right\} \right),$$

- $U_k(h, i_k)$ is an upper bound on the probability that arm i_k is pulled at time k given the history of pulls (according to h) up to time $k-1$:

$$U_k(h, i_k) = \begin{cases} \int_0^1 \left(\frac{p}{m_k} \mathbb{1}_{\{b_k = 1\}} + (1-p) u_k(h, i_k) \mathbb{1}_{\{b_k = 0\}} \right) f_{M(h,k)}(g(p)) \, \mathrm{d}p & \text{if } k < t \\ \int_0^1 \left(\frac{p}{m_t} + (1-p) u_t(h, j) \right) f_{M(h,t)}(g(p)) \, \mathrm{d}p & \text{if } k = t \end{cases} \tag{5}$$

- $U_t(h,j)$ is an upper bound on the probability that arm j was pulled at time t given the history of pulls (according to h) up to time $t-1$:

$$U_t(h,j) = \int_0^1 \left(\frac{p}{m_t} + (1-p)u_t(h,j) \right) f_{M(h,t)}(g(p))\, \mathrm{d}p. \tag{6}$$

In standard regret bounds, the bound is usually in terms of the mean rewards μ_j and Δ_j for each arm j, which are not known in the application. Our bounds analogously depend on the μ_j's and $\Delta(i,i_s)$'s (where i_s is the arm played at time s, and i is another arm with higher mean reward). While standard bounds usually have a simple dependence on μ_j's, our bounds have a more complicated dependence on the μ_j's. On the other hand, they depend on the same quantities as the standard bounds, once we have the μ_j terms, the bound can be computed using the same information that is available in the standard bounds. For instance $u_s(h,i_s)$, $U_s(h,i_s)$, and $U_t(h,j)$ do not require any additional information other than the μ_j's. Theorem 2.1 presents a finite time upper bound on the regret for the AG-L algorithm (see Appendix A for the proof).

Theorem 2.1. *The bound on the mean regret $\mathbb{E}[R_n]$ at time n is given by*

$$\mathbb{E}[R_n] \leq \sum_{j \in M_I} \Delta_{j,i_t^*} + \sum_{t=m_I+1}^{n} \sum_{j \in M_t(\mathcal{L})} \Delta_{j,i_t^*} \sum_{h \in \mathcal{H}_{t-1}} \left(U_t(h,j) \prod_{s=m_I+1}^{t-1} U_s(h,i_s) \right). \tag{7}$$

The first summation in (7) represents the total regret suffered during the initialization phase. Intuitively, we can explain why this regret bound is better than the one that you would get by using the standard AG policy by looking at the quantities in Equation (5). The integrand has two main terms: $1/m_s$ (recall that m_s is the number of arms available at turn s) and $u_s(h,i_s)$. The first is a constant that appears during exploration phases (when $b_s = 1$), while the second one is a product of negative exponentials that gets quickly smaller than $1/m_s$ given enough pulls on arm i_s and it appears during exploitation turns (when $b_s = 0$). If the algorithm uses arms that are bound to expire quickly (for example using the AG policy), it will explore more often, thus having the $1/m_s$ term appear more often than $u_s(h,i_s)$ term, yielding a worse regret. Conversely, by considering only the set $M_t(\mathcal{L})$ of arms with long life, the term $1/m_s$ will appear less often than the smaller quantity $u_s(h,i_s)$.
We can see this by restating Theorem 2.1 as

Theorem 2.2. *The bound on the mean regret $\mathbb{E}[R_n]$ at time n is given by*

$$\mathbb{E}[R_n] \leq \mathcal{O}(1) + \sum_{t=m_I+1}^{n} \sum_{j \in M_t(\mathcal{L})} \Delta_{j,i_t^*} \tag{8}$$

$$\times \sum_{h \in \mathcal{H}_{t-1}} F\left(\frac{1}{m_1}, \cdots, \frac{1}{m_t}, u_1(h,i_1), \cdots, u_t(h,i_t) \right), \tag{9}$$

where,

$$F\left(\frac{1}{m_1}, \cdots, \frac{1}{m_t}, u_1(h,i_1), \cdots, u_t(h,i_t) \right) = U_t(h,j) \prod_{s=m_I+1}^{t-1} U_s(h,i_s)$$

is an increasing function in all its arguments. Intuitively, $u_1(h,i_1), \cdots, u_t(h,i_t)$ are smaller than $\frac{1}{m_1}, \cdots, \frac{1}{m_t}$ since they decrease at a fast rate (they are products of negative exponentials). By regulating exploration on arms that live longer, the bound of Algorithm 1 presents more times the smaller terms than the higher ones, yielding an overall better expected regret.

Another way of seeing why it is beneficial to limit exploration only among arms with long life is to think about two different games: the first with 100 arms, and the second with 10 arms. The quality of the arms come from the same distribution, for example we know that 30% of the arms have high expected rewards, and 70% have instead low expected rewards. The probability of picking a bad arm is the same in both games, but it is much easier to realize the algorithm is playing a bad arm when you only have 10 arms available, since you can allocate more pulls to the same arm to build a more accurate mean estimate. So you will explore less (the term $1/m_s$ will appear less often) and have better bounds on the probability of playing suboptimal arms (the terms $u_s(h,i_s)$ decrease more quickly). In short, what Algorithm 1 is achieving, is to reduce the initial game to an easier one where the arms with long life represent the entire set of arms but will yield to less turns allocated to exploration.

2.2 The UCB mortal with life regulation algorithm

Algorithm 2 introduces a modified version of the UCB algorithm of Auer et al. [2002]. In the standard UCB algorithm, the arm with the highest upper confidence bound above the estimated mean is played. In this new version, the upper confidence

bound has been modified so that it can be used in the mortal setting, and it gradually shrinks on the estimated mean as the life of the arm decreases. In this way, arms that are close to expiring are played only if their estimated mean is high, and exploration is encouraged only on arms that have a long lifespan. Let s_j and l_j be the first and last turn at which arm j is available, and let $\psi(j,t)$ a function proportional to the remaining life of arm j decreases. An example for $\psi(j,t)$ is $c\log(l_j - t + 1)$, where c is a positive constant (note that $\psi(j,t)$ approaches to zero as the game gets closer to the expiration turn of arm j). New arms are initialized by using the average performance of past arms (i.e., if in the past a lot of bad arms appeared, new arms are considered more likely to be bad, and vice-versa if lots of good arms appeared in the past) and their upper confidence bound is build as if they have been played once. We abbreviate this algorithm with UCB-L.

Algorithm 2: UCB-L algorithm

Input : number of rounds n, initialization set of arms M_I,, set M_t of available arms at time, rewards range $[a,b]$

Initialization: play all arms in M_I once, and initialize $\widehat{X}_j$ for each $j = 1, \cdots, m_I$

for $t = m_I + 1$ **to** n **do**

Play arm with highest

$$\widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}};$$

Get reward $X_j(t)$;

Update $\widehat{X}_j$;

end

Theorem 2.3 presents a finite time regret bound for the UCB-L algorithm (proof in Appendix B).

Theorem 2.3. *Let $\bigcup_{z=1}^{E_j} L_j^z$ be a partition of L_j into epochs with different best available arm, s_j^z and l_j^z be the first and last step of epoch L_j^z, and for each epoch let $u_{j,z}$ be defined as*

$$u_{j,z} = \max_{t\in\{s_j^z,\cdots,l_j^z\}} \left\lceil \frac{8\psi(j,t)\log(t-s_j)}{\Delta_{j,z}^2} \right\rceil,$$

where

$$\Delta_{j,i_t^*} = \Delta_{j,z} \ \textit{for}\ t \in L_j^z.$$

Then, the bound on the mean regret $\mathbb{E}[R_n]$ at time n is given by

$$\begin{aligned}\mathbb{E}[R_n] \quad &\leq \quad \sum_{j\in M_I} \Delta_{j,i_t^*} \\ &+ \quad \sum_{j\in M}\sum_{z=1}^{E_j} \Delta_{j,z} \min\left(l_j^z - s_j^z \ , \ u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} (t-s_{i_t^*})(t-s_j-u_{j,z}+1)\left[(t-s_j)^{-\frac{4}{r^2}\psi(j,t)} + (t-s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)}\right]\right).\end{aligned}$$

The first summation $\sum_{j\in M_I} \Delta_{j,i_t^*}$ is the regret suffered during the initialization phase. Intuitively, $u_{j,z}$ is the number of pulls required to be able to distinguish arm j from the best arm in epoch z, while the quantity in the last summation is an upper bound on the probability that we are either underestimating the best arm in epoch z or we are overestimating arm j. If the game is such that no new arms are born during the game and all arms expire after turn n, then this regret bound reduces to the standard UCB bound (see Auer et al. [2002]).

Algorithm 5 is a similar adaptation of the LinUCB algorithm introduced by Li et al. [2010] to the mortal setting. Also in this case, the function $\psi(j,t)$ regulates the amplitude of the upper confidence bound above the estimated mean according to the remaining life of the arm. As before, new arms are initialized by using the average performance of past arms (i.e., if in the past a lot of bad arms appeared, new arms are considered more likely to be bad, and vice-versa if lots of good arms appeared in the past).

Algorithm 3: LinUCB-L algorithm

Input : number of rounds n, initial set of arms M_I, set M_t of available arms at time, rewards range $[a, b]$, dimension d (context space dimension + arms space dimension)

Initialization: For each $j \in M_I$, $A_j = I_d$, $b_j = 0_{d\times 1}$

for $t = 1$ **to** n **do**

  Get context x_t (or $x_{t,j}$ if each arm gets its context);

  for $j = 1$ **to** m_t **do**

    Set $\hat{\theta}_j = A_j^{-1} b_j$;

    Set $UCB_j = \hat{\theta}_j^T x_t + \psi(j,t)\sqrt{x_t^T A_j^{-1} x_t}$;

  end

  Play arm $j = \text{argmax}_i\, UCB_i$;

  Get reward $X_j(t)$;

  Update $A_j = A_j^{-1} + x_t x_t^T$;

  Update $b_j = b_j + r_t x_t$;

end

3 Experimental results: importance of discarding arms with low lifespan.

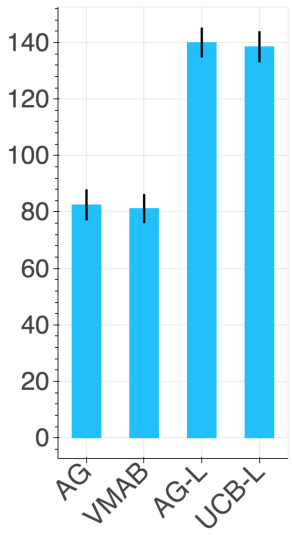
In the following we compare the average of final cumulative rewards for each algorithm over 50 games with 1000 turns in each game. Rewards are drawn from Bernoulli distributions and new arms arrive according to a Poisson process with parameter 1.1. The algorithms are:

- AG: adaptive greedy algorithm from Chakrabarti et al. [2009];
- VMAB: volatile UCB algorithm from Bnaya et al. [2013];
- AG-L: adaptive greedy algorithm with life regulation (Algorithm 1);
- AG-L(predict): adaptive greedy algorithm with life regulation, where remaining lifespan is predicted from arm features;
- UCB-L: UCB mortal algorithm with life regulation (Algorithm 2);
- UCB-L(predict): UCB mortal algorithm with life regulation, where remaining lifespan is predicted from arm features;
- LinUCB-L: LinUCB contextual mortal algorithm with life regulation (Algorithm 5);
- LinUCB-L(predict): LinUCB contextual mortal algorithm with life regulation, where remaining lifespan is predicted from arm features.

3.1 Two types of arms with short and long life span.

In Figure 1 we show the average of final cumulative rewards in the case where the generated arms are born from the same reward distribution (that is, their mean reward, or parameter of the Bernoulli distribution, is chosen uniformly between $[0, 1]$) but their lifespans are substantially different. With probability 0.3 the arm's lifespan is drawn from a Poisson distribution with mean 250, while with probability 0.7 the arm's lifespan is drawn from a Poisson distribution with mean 60.

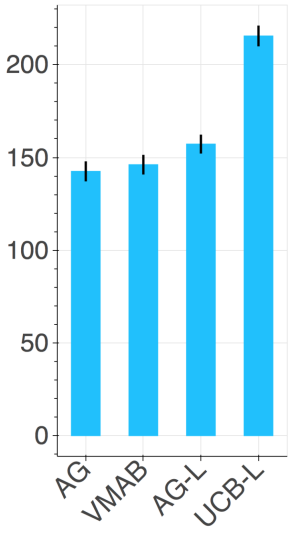
Figure 1



3.2 One type of arm with medium life span.

In Figure 2 we show the average of final cumulative rewards in the case where the generated arms are born from the same reward distribution (that is, their mean reward, or parameter of the Bernoulli distribution, is chosen uniformly between $[0, 1]$) and their lifespan is drawn from a Poisson distribution with mean 100. In this case, the AG-L algorithm is not as effective as before, and performs almost like the AG algorithm. One flaw common to the adaptive greedy algorithms is that they compute the probability of exploration based on the best performing arm available. That means that if the best arm available has not a mean reward close to the maximum of the range of rewards $[a, b]$, then the algorithm will explore anyway (for example, if the best arm available has a parameter of its Bernoulli reward distribution of 0.5, the algorithm has a 50% chance to explore even if it has detected the best arm).

Figure 2

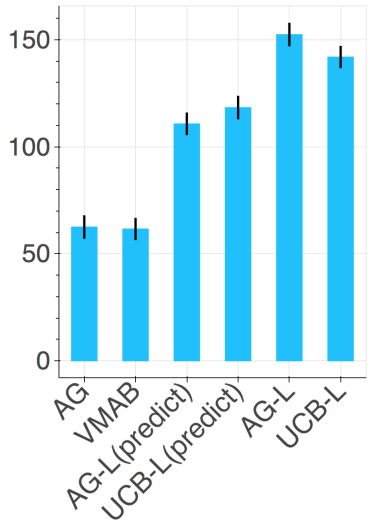


3.3 Two types of arms with features having short and long life span.

In Figure 3 we show the average of final cumulative rewards in the case where the generated arms are born from the same reward distribution (that is, their mean reward, or parameter of the Bernoulli distribution, is chosen uniformly between $[0, 1]$) but their lifespans can be substantially different and are decided from features of the arms. In this case, we used the value of three normally distributed r.v. with mean 30 and standard deviation 15 as arm features. Algorithms AG-L(predict) and

UCB-L(predict) use the features to predict the value of the remaining lifespan based on the data gathered by the arms that have died. The other algorithms know the exact lifespan of the arms and thus have an advantage.

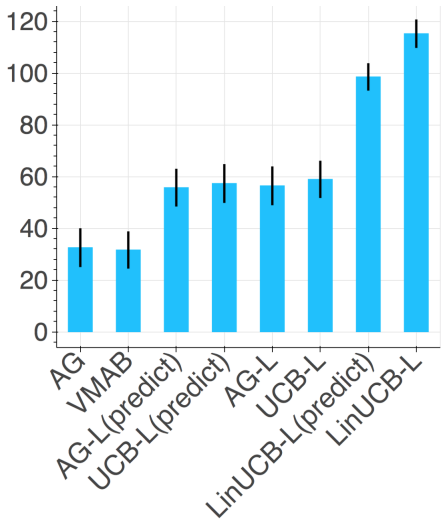
Figure 3



3.4 Contextual problem with two types of arms with features having short and long life span.

In Figure 4 we show the average of final cumulative rewards in the case where the rewards depend on a context shown at each turn (in this case, the context consists of a vector with three entries generated from normal random variables) as well as the features of the arms discussed earlier. The lifespans of the arms are decided from features of the arms as described in Section 3.3. Algorithms AG-L(predict), UCB-L(predict), and LinUCB-L use the features to predict the value of the remaining lifespan based on the data gathered by the arms that have died. The other algorithms know the exact lifespan of the arms. Only LinUCB-L and LinUCB-L(predict) are designed to work in a contextual setting, but we included the other algorithms as well since the contextual space is small in this case.

Figure 4

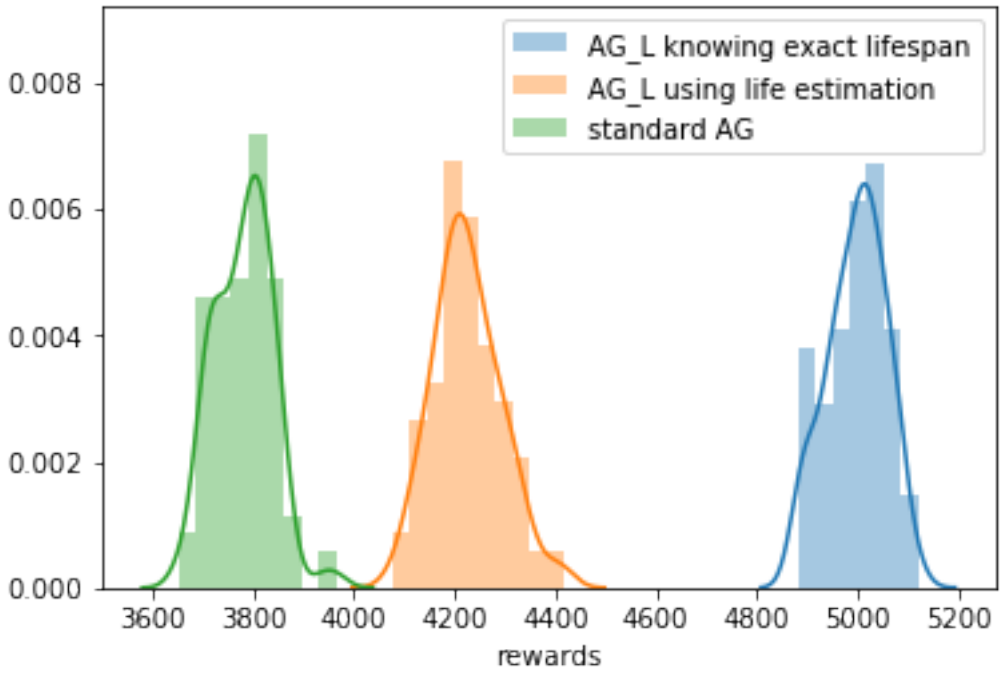


We simulated AG-L/UCB-L and the standard AG/UCB using the dataset from the Yahoo Webscope program. The dataset consists of a stream of recommendation events that display articles randomly to users. Each event contains information of the action taken, the outcome of that action, the candidate arm pool at that time and the associated timestamp. We preprocessed the original text file into structured data frame **??**.

Table 1: extracted dataframe from the original text record

timestamp	id	clicked	number of candidates
1317513291	id-560620	0	26
1317513291	id-565648	0	26
1317513291	id-563115	0	26
1317513292	id-552077	0	26
1317513292	id-564335	0	26

Figure 5: AGs playing ”the same” game 100 times



At each game, each simulator is fed a recommender algorithm and a portion of the dataset that starts at the same time. The recommender starts playing from that time point until a fixed number of turns are played. Then the simulators record the accumulated reward of its recommender as the measurement for its performance.

As the recommenders play along and make their decisions, our offline simulator will only consider an event as valid if the displayed article in the event matches the choice of the recommender (we do not know the outcome of actions not recorded in the dataset, so we treat those tempted but not happening events as invalid). Therefore, while still playing the same number of turns, a recommender in the ”unlucky” simulator that discards a lot of events might live longer than other recommenders. But overall this difference is small. More details on the experiment can be found in the Appendix.

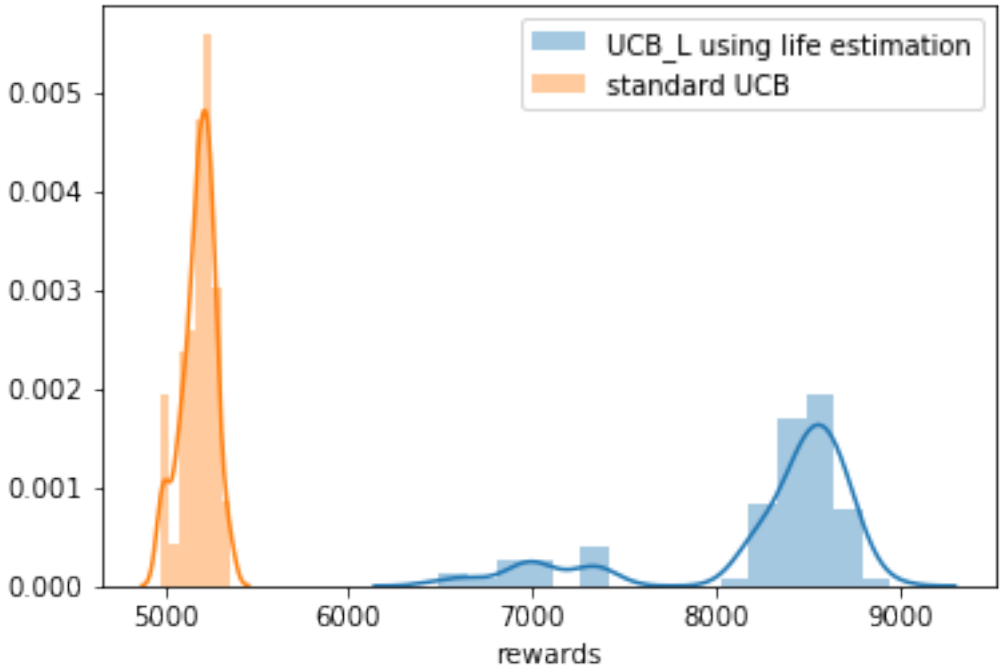
The original AG-L and UCB-L presume we know at what time an article appears and expires. We can pre-scan all the records to get information of each article's lifespan. In real life applications, however, we don't usually have this information. So, we can estimate new arms' lifespan using the mean lifespan of expired arms. In the experiment we also included such recommenders that use estimated lifespan.

In order to obtain a distribution for performance (rather than a single performance measurement), we reran the AGs many times to plot the histogram – they are non-deterministic as they choose arms randomly from candidates with enough remaining life. For UCBs, however, they are deterministic because they always pick the arm with the best upper confidence bound. Reruning UCBs with the same dataset will always give the same result. Therefore, we reran the UCBs for different sliding windows of time. That is, we started the algorithms at many different points in time.

Figure **??** and **??** show the histogram of rewards for different strategies. Each recommender played 100 games with 100000 turns each game, consuming millions of data rows each game. We can see recommenders with life regulation outperform the standard ones most of the time. Among AG-Ls, knowing the exact lifespan of each article (rather than using an estimated lifespan) improves the performance.

We could try to apply the same sliding window trick to AGs so that we can plot AGs' result together with UCBs'. However,

Figure 6: UCBs playing ”the same” set of 100 different games



in the experiment UCBs' simulator tend to be more unlucky so they discard more records and live longer than AGs. Articles in later records might be more popular, thus the recommender algorithm might easily gain reward when its simulator come to this ”fruitful” part of record. Then, this comparison would be unfair to AGs. This is why we didn't include this comparison.

4 Conclusions

In this work, we have shown that it is possible to leverage knowledge about the lifetimes of the arms to improve the quality of exploration and exploitation in mortal multi-armed bandits. Our algorithms focus on exploring the arms that will be available longer, leading to substantially increased rewards. This is true also for contextual mortal multi-armed bandits where each arm has features, and where context also changes over time. In cases where we do not know the lifetimes of the arms but can estimate them, these techniques are still able to substantially increase rewards. These techniques were inspired by a high-scoring entry of the Exploration-Exploitation 3 competition where this strategy led to a substantial improvement in score.

References

Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Explore/exploit schemes for web content optimization. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 1–10. IEEE, 2009. 1

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. 1, 2.2, 2.2, B, B, C

Zahy Bnaya, Rami Puzis, Roni Stern, and Ariel Felner. Volatile multi-armed bandits for guaranteed targeted social crawling. In *AAAI (Late-Breaking Developments)*, 2013. 1, 3

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Advances in neural information processing systems*, pages 273–280, 2009. 1, 2.1, 3

R Feraud, T Urvoy ACML, and 2012. A stochastic bandit algorithm for scratch games. *jmlr.org*. 1

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. 2.2, D.6

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 297–306, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0493-1. doi: 10.1145/1935826.1935878. URL `http://doi.acm.org/10.1145/1935826.1935878`. D.2

Setareh Maghsudi and Slawomir Stańczak. On channel selection for energy-constrained rateless-coded d2d communications. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 1028–1032. IEEE, 2015. 1

R. J. Vaughan and W. N. Venables. Permanent expressions for order statistic densities. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):308–310, 1972. 2.1, A

A The regret bound of the Adaptive greedy algorithm

We present a finite-time bound on the cumulative regret defined in Equation (2).
Let $\mathcal{H}_{t-1}$ is the set of all possible histories (after deterministic initialization) of the game up to turn $t-1$:

$$\mathcal{H}_{t-1} = \left\{ h = \begin{bmatrix} b_{m_I+1} & b_{m_I+2} & \dots & b_{t-1} \\ i_{m_I+1} & i_{m_I+2} & \dots & i_{t-1} \end{bmatrix} : b_s \in \{0,1\},\ i_s \in M_s, \quad \forall s \in \{m_I+1, \dots, t-1\} \right\}. \tag{10}$$

If $b_s = 1$ we say that the algorithm explored at time s, if $b_s = 0$ we say that the algorithm exploited at time s, while i_s is the index of the arm that was played at time s.

Theorem 2.1 *Let us define the following quantities:*

- $g(p) = b + (a-b)p$,
- $f_{M(h,s)}(g(p))$ *is the PDF (or PMF) of the maximum of the estimated mean rewards at time* s *given that each arm has been pulled according to history* h *up to time* $s-1$*:*

$$f_{M(h,k)}(x) = \frac{1}{(m_t-1)!} \operatorname{perm} \left(\begin{bmatrix} F_1(x) & F_2(x) & \dots & F_{m_k}(x) \\ \vdots & \vdots & \vdots & \vdots \\ F_1(x) & F_2(x) & \dots & F_{m_k}(x) \\ f_1(x) & f_2(x) & \dots & f_{m_k}(x) \end{bmatrix} \right) \Bigg\} \ m_k - 1 \text{ rows} \ ,$$

  where $f_1(x), \cdots, f_{m_k}(x)$ *and* $F_1(x), \cdots, F_{m_k}(x)$ *are the PDFs (or PMFs) of the distributions of the average rewards,*

- $u_s(h, i_s)$ *is an upper bound on the probability that arm* i_s *is considered to be the best arm at time* s *given the history of pulls (according to* h*) up to time* $s-1$*:*

$$u_s(h, i_s) = \prod_{i:\mu_i > \mu_{i_s}} \left(\exp\left\{ -\frac{t_{i_s}(h,s)\Delta(i,i_s)^2}{2r} \right\} + \exp\left\{ -\frac{t_i(h,s)\Delta(i,i_s)^2}{2r} \right\} \right),$$

- $U_s(h, i_s)$ *is an upper bound on the probability that arm* i_s *was pulled at time* s *given the history of pulls (according to* h*) up to time* $s-1$*:*

$$U_s(h, i_s) = \int_0^1 \left(\frac{p}{m_s} \mathbb{1}_{\{b_s=1\}} + (1-p) u_s(h, i_s) \mathbb{1}_{\{b_s=0\}} \right) f_{M(h,s)}(g(p))\, dp,$$

- $u_t(h, j)$ *is an upper bound on the probability that arm* j *is considered to be the best arm at time* t *given the history of pulls (according to* h*) up to time* $t-1$*:*

$$u_t(h, j) = \prod_{i:\mu_i > \mu_j} \left(\exp\left\{ -\frac{t_j(h,t)\Delta(i,j)^2}{2r} \right\} + \exp\left\{ -\frac{t_i(h,t)\Delta(i,j)^2}{2r} \right\} \right),$$

- $U_t(h, j)$ *is an upper bound on the probability that arm* j *was pulled at time* t *given the history of pulls (according to* h*) up to time* $t-1$*:*

$$U_t(h, j) = \int_0^1 \left(\frac{p}{m_t} + (1-p) u_t(h, j) \right) f_{M(h,t)}(g(p))\, dp.$$

Then, an upper bound on the expected cumulative regret R_n *at round* n *is given by*

$$\mathbb{E}[R_n] \leq \sum_{j \in M_I} \Delta_{j,i_j^*} + \sum_{t=m_I+1}^{n} \sum_{j \in M_t} \Delta_{j,i_t^*} \sum_{h \in \mathcal{H}_{t-1}} \left(U_t(h, j) \prod_{s=m_I+1}^{t-1} U_s(h, i_s) \right).$$

First step: Decomposition of $\mathbb{E}[R_n]$.

$$\mathbb{E}[R_n] = \sum_{j \in M_I} \Delta_{j,i_j^*} + \sum_{t=m_I+1}^{n} \sum_{j \in M_t} \Delta_{j,i_t^*} \mathbb{P}\left(t \in I(j)\right), \tag{11}$$

where we can write $\mathbb{P}\left(t \in I(j)\right)$ as

$$\mathbb{P}\left(t \in I(j)\right) = \sum_{h \in \mathcal{H}_{t-1}} \mathbb{P}\left(t \in I(j) \;\middle|\; H_{t-1} = h\right) \mathbb{P}\left(H_{t-1} = h\right), \tag{12}$$

where H_{t-1} is a random variable that takes values in $\mathcal{H}_{t-1}$ defined as

$$\mathcal{H}_{t-1} = \left\{ h = \begin{bmatrix} b_{m_I+1} & b_{m_I+2} & \dots & b_{t-1} \\ i_{m_I+1} & i_{m_I+2} & \dots & i_{t-1} \end{bmatrix} : b_s \in \{0,1\},\ i_s \in M_s \ \ \forall s \in \{m_I+1, \dots, t-1\} \right\}. \tag{13}$$

$\mathcal{H}_{t-1}$ is the set of all possible histories (after deterministic initialization) of the game up to turn $t-1$. If $b_s = 1$ we say that the algorithm explored at time s, if $b_s = 0$ we say that the algorithm exploited at time s, while i_s is the index of the arm that was played at time s. The set $\mathcal{H}_{t-1}$ has $\prod_{s=m_I+1}^{t-1}(2m_s)$ elements. Note also that, by design of the algorithm, if an arm j is new at time s,

$$\mathbb{P}\left(H_{t-1} = \begin{bmatrix} b_{m_I+1} & \dots & b_s = 0 & \dots & b_{t-1} \\ i_{m_I+1} & \dots & i_s = j & \dots & i_{t-1} \end{bmatrix}\right) = 0,$$

because the algorithm does not allow exploitation of a new arm. In the following steps we study (and find an upper bound when needed) each term in (12).

Second step: Upper bound for $\mathbb{P}\left(H_{t-1} = h\right)$.

Let us define h_k, with $k > m_I,\ k \in \mathbb{N}$, the first $k - m_I$ columns of h (so h and h_{t-1} are the same).
For each h, we indicate how many times arm j has been pulled up to time k with

$$t_j(h,k) = \mathbb{1}_{\{j \in M_I\}} + \sum_{s=m_I+1}^{k} \mathbb{1}_{\{i_s \in I(j)\}},$$

and, similarly to the definition of $\widehat{X}_j$ given in (1), we denote the mean estimated reward for arm j, given history of pulled arms h, with

$$\widehat{X}_j(h,k) = \frac{1}{t_j(h,k-1)} \sum_{s \in I(j)}^{t_j(h,k-1)} X_j(s). \tag{14}$$

For each h, the probability of exploration at time k is a random variable $E(h,k)$ with distribution given by

$$\mathbb{P}\left(E(h,k) = p\right) = \mathbb{P}\left(1 - \frac{\max_{j \in M_k} \widehat{X}_j(h,k) - a}{b-a} = p\right), \tag{15}$$

Let us define $g(p) = b + (a-b)p$, then we can rewrite (15) as

$$\mathbb{P}\left(E(h,k) = p\right) = \mathbb{P}\left(\max_{j \in M_k} \widehat{X}_j(h,k) = g(p)\right). \tag{16}$$

We will give a formula for 16 in the next step of the proof.

We can compute $\mathbb{P}\left(H_{t-1} = h\right)$ recursively using the fact that $\mathbb{P}\left(H_{t-1} = h\right)$ is equal to

$$\mathbb{P}\left(H_{t-1} = h \,\middle|\, H_{t-2} = h_{t-2}\right) \mathbb{P}\left(H_{t-2} = h_{t-2} \,\middle|\, H_{t-3} = h_{t-3}\right) \cdots \mathbb{P}\left(H_{m_I+2} = h_{m_I+2} \,\middle|\, H_{m_I+1} = h_{m_I+1}\right) \mathbb{P}\left(H_{m_I+1} = h_{m_I+1}\right). \tag{17}$$

h_{m_I+1} has only one column: $\begin{bmatrix} b_{m_I+1} \\ i_{m_I+1} \end{bmatrix}$, where $b_{m_I+1} \in \{0,1\}$ and $i_{m_I+1} \in M_{m_I+1}$.

We can write $\mathbb{P}\left(H_{m_I+1} = h_{m_I+1}\right)$ as

$$\int_0^1 \left(\frac{p}{m_{I+1}} \mathbb{1}_{\{b_{m_I+1}=1\}} + (1-p)\mathbb{P}\left(\widehat{X}_{i_{m_I+1}}(h, m_I+1) > \widehat{X}_i(h, m_I+1) \ \ \forall i \neq i_{m_I+1}\right) \mathbb{1}_{\{b_{m_I+1}=0\}} \right) \mathbb{P}\left(E(h, m_I+1) = p\right) \, \mathrm{d}p \tag{18}$$

Similarly, we can compute each term in (17). For each $s \in \{m_I+2, \cdots, t-1\}$, we have that $\mathbb{P}\left(H_s = h \,\middle|\, H_{s-1} = h_{s-1}\right)$ is given by

$$\int_0^1 \left(\frac{p}{m_s}\mathbb{1}_{\{b_s=1\}} + (1-p)\mathbb{P}\left(\widehat{X}_{i_s}(h,s) > \widehat{X}_i(h,s) \;\; \forall i \neq i_s\right)\mathbb{1}_{\{b_s=0\}}\right)\mathbb{P}\left(E(h,s)=p\right)\,\mathrm{d}p \tag{19}$$

Using independence of the arms and Proposition 3, for each $s \in \{m_I+1, \cdots, t-1\}$ we can write

$$\begin{aligned}
& \mathbb{P}\left(\widehat{X}_{i_s}(h,s) > \widehat{X}_i(h,s) \;\; \forall i \neq i_s\right) && (20)\\
\leq \;& \mathbb{P}\left(\widehat{X}_{i_s}(h,s) > \widehat{X}_i(h,s) \;\; \forall i : \mu_i > \mu_{i_s}\right) && (21)\\
\leq \;& \prod_{i:\mu_i>\mu_{i_s}} \mathbb{P}\left(\widehat{X}_{i_s}(h,s) > \widehat{X}_i(h,s)\right) && (22)\\
\leq \;& \prod_{i:\mu_i>\mu_{i_s}} \left[\mathbb{P}\left(\widehat{X}_{i_s}(h,s) > \mu_{i_s} + \frac{\Delta(i,i_s)}{2}\right) + \mathbb{P}\left(\widehat{X}_i(h,s) < \mu_i - \frac{\Delta(i,i_s)}{2}\right)\right] && (23)
\end{aligned}$$

and then bound each term by using Hoeffding's inequality[1]:

$$\mathbb{P}\left(\widehat{X}_{i_s}(h,s) > \mu_{i_s} + \frac{\Delta(i,i_s)}{2}\right) \leq \exp\left\{-\frac{t_{i_s}(h,s)\Delta(i,i_s)^2}{2r}\right\} \tag{24}$$

and

$$\mathbb{P}\left(\widehat{X}_i(h,s) < \mu_i - \frac{\Delta(i,i_s)}{2}\right) \leq \exp\left\{-\frac{t_i(h,s)\Delta(i,i_s)^2}{2r}\right\}. \tag{25}$$

Let us define

$$u_s(h,i_s) = \prod_{i:\mu_i>\mu_{i_s}} \left(\exp\left\{-\frac{t_{i_s}(h,s)\Delta(i,i_s)^2}{2r}\right\} + \exp\left\{-\frac{t_i(h,s)\Delta(i,i_s)^2}{2r}\right\}\right), \tag{26}$$

then, $\mathbb{P}\left(H_s = h \,\middle|\, H_{s-1} = h_{s-1}\right) \leq U_s(h,i_s)$, where

$$U_s(h,i_s) = \int_0^1 \left(\frac{p}{m_s}\mathbb{1}_{\{b_s=1\}} + (1-p)u_s(h,i_s)\mathbb{1}_{\{b_s=0\}}\right)\mathbb{P}\left(E(h,s)=p\right)\,\mathrm{d}p, \tag{27}$$

and from (17)

$$\mathbb{P}\left(H_{t-1} = h\right) \leq \prod_{s=m_I+1}^{t-1} U_s(h,i_s). \tag{28}$$

Third step: Formula for $\mathbb{P}\left(E(h,k)=p\right)$.

We can determine $\mathbb{P}\left(E(h,k)=p\right) = \mathbb{P}\left(\max_{j\in M_k} \widehat{X}_j(h,k) = g(p)\right)$ by using a result from Vaughan and Venables [1972] that describes the PDF of the maximum of random variables coming from different distributions. Note that each $\widehat{X}_j(h,k)$ has a different distribution[2] that depends also on s_j. Given a square matrix A, let $\mathrm{perm}(A)$ be the permanent[3] of A. Then, the PDF of $\max_{j\in M_t} \widehat{X}_j(h,k)$ is given by

$$f_{M(h,k)}(x) = \frac{1}{(m_t-1)!}\,\mathrm{perm}\left(\begin{bmatrix} F_1(x) & F_2(x) & \dots & F_{m_k}(x) \\ \vdots & \vdots & \vdots & \vdots \\ F_1(x) & F_2(x) & \dots & F_{m_k}(x) \\ f_1(x) & f_2(x) & \dots & f_{m_k}(x) \end{bmatrix}\right) \;\Bigg\}\; m_k - 1 \text{ rows} \tag{29}$$

[1] **Hoeffding's bound:** Let $X_1, \cdots, X_n$ be r.v. bounded in $[a_i, b_i]$ $\forall i$. Let $\widehat{X} = \frac{1}{n}\sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[\widehat{X}]$. Then, $\mathbb{P}\left(\widehat{X} - \mu \geq \varepsilon\right) \leq \exp\left\{-\frac{2n^2\varepsilon^2}{\sum_{i=1}^n (b_i-a_i)^2}\right\}$. In our case, $\varepsilon = \frac{\Delta_{i_s,i}}{2}$, $n = t_s$ or t_{i_s}, $b-a = r$.

[2] For example, if X_i has a Bernoulli distribution with parameter μ_i, $\widehat{X}_{i,2}$ assumes values in $\{0,1,2\}$, with probabilities $(1-\mu_i)^2, (1-\mu_i)\mu_i, \mu_i^2$, while $\widehat{X}_{i,3}$ assumes values in $\{0,1,2,3\}$, with probabilities $(1-\mu_i)^3, (1-\mu_i)^2\mu_i, (1-\mu_i)\mu_i^2, \mu_i^3$.

[3] The permanent of a square matrix A is defined like the determinant, except that all signs are positive.

where $f_1(x), \cdots, f_{m_k}(x)$ and $F_1(x), \cdots, F_{m_k}(x)$ are the PDFs (or PMFs) of the cumulative distributions of the average rewards $\widehat{X}_j(h,k)$ of arms $j \in M_k$ (if unknown, they are approximated by a Normal r.v. by CLT). Thus,

$$\mathbb{P}\left(E(h,k) = p\right) = f_{M(h,k)}(g(p)). \tag{30}$$

Fourth step: Formula for $\mathbb{P}\left(t \in I(j) \;\middle|\; H_{t-1} = h\right)$.

We have that

$$\mathbb{P}\left(t \in I(j) \;\middle|\; H_{t-1} = h\right) = \int_0^1 \left[p\frac{1}{m_t} + (1-p)\mathbb{P}\left(\widehat{X}_j(h,t) > \widehat{X}_i(h,t) \;\; \forall i \neq j\right)\right] f_{M(h,t)}(g(p))\mathrm{d}p \tag{31}$$

Similarly to Step 2, $\mathbb{P}\left(\widehat{X}_j(h,t) > \widehat{X}_i(h,t) \;\; \forall i \neq j\right)$ has upper bound

$$u_t(h,j) = \prod_{i:\mu_i > \mu_j} \left(\exp\left\{-\frac{t_j(h,t)\Delta(i,j)^2}{2r}\right\} + \exp\left\{-\frac{t_i(h,t)\Delta(i,j)^2}{2r}\right\}\right), \tag{32}$$

and (31) has upper bound $U_t(h,j)$, where

$$U_t(h,j) = \int_0^1 \left(\frac{p}{m_t} + (1-p)u_t(h,j)\right) f_{M(h,t)}(g(p)) \; \mathrm{d}p. \tag{33}$$

Note that $U_t(h,j)$ is different from $U_s(h,i_s)$ defined in (27) that have values of b_s available.

Fifth step: Bringing together all the bounds of the previous steps.

From (12) we have that

$$\mathbb{P}\left(t \in I(j)\right) = \sum_{h \in \mathcal{H}_{t-1}} \mathbb{P}\left(t \in I(j) \;\middle|\; H_{t-1} = h\right) \mathbb{P}\left(H_{t-1} = h\right) \leq \sum_{h \in \mathcal{H}_{t-1}} U_t(h,j) \prod_{s=m_I+1}^{t-1} U_s(h,i_s), \tag{34}$$

and from (11) in conclusion:

$$\mathbb{E}[R_n] \leq \sum_{j \in M_I} \Delta_{j,i_j^*} + \sum_{t=m_I+1}^{n} \sum_{j \in M_t} \Delta_{j,i_t^*} \sum_{h \in \mathcal{H}_{t-1}} \left(U_t(h,j) \prod_{s=m_I+1}^{t-1} U_s(h,i_s)\right).$$

B The regret bound of the UCB mortal algorithm

Theorem 2.3 *Let $\bigcup_{z=1}^{E_j} L_j^z$ be a partition of L_j into epochs with different best available arm, s_j^z and l_j^z be the first and last step of epoch L_j^z, and for each epoch let $u_{j,z}$ be defined as*

$$u_{j,z} = \max_{t \in \{s_j^z, \cdots, l_j^z\}} \left\lceil \frac{8\psi(j,t)\log(t-s_j)}{\Delta_{j,z}^2} \right\rceil, \tag{35}$$

where

$$\Delta_{j,i_t^*} = \Delta_{j,z} \ \text{ for } t \in L_j^z. \tag{36}$$

Then, the bound on the mean regret $\mathbb{E}[R_n]$ at time n is given by

$$\begin{aligned}\mathbb{E}[R_n] \quad &\leq \quad \sum_{j \in M_I} \Delta_{j,i_j^*} \\ &+ \quad \sum_{j \in M} \sum_{z=1}^{E_j} \Delta_{j,z} \min \left(l_j^z - s_j^z \ , \ u_{j,z} + \sum_{\substack{t \in L_j^z \\ t > m_I}} (t - s_{i_t^*})(t - s_j - u_{j,z} + 1) \left[(t-s_j)^{-\frac{4}{r^2}\psi(j,t)} + (t - s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)} \right] \right).\end{aligned}$$

First step: Decomposition of $\mathbb{E}[R_n]$.

Let us partition the set of steps L_j during which arm j is available into E_j epochs L_j^z, such that

- $\bigcup_{z=1}^{E_j} L_j^z = L_j$,
- $L_j^{z_1} \cap L_j^{z_2} = \emptyset$ if $z_1 \neq z_2$,
- $i_t^* \neq i_s^*$ if $t \in L_j^{z_1}$ and $s \in L_j^{z_2}$ (i.e., if different epochs have different best arm available).

Since during the same epoch the best arm available does not change, let us define

$$\Delta_{j,i_t^*} = \Delta_{j,z} \ \text{ for } t \in L_j^z, \tag{37}$$

and $s_j^z = \min L_j^z$, $l_j^z = \max L_j^z$ the first and last step of epoch L_j^z.
Then, using the second formulation of the cumulative regret given in (3) we have that

$$R_n \quad = \quad \sum_{j \in M_I} \Delta_{j,i_j^*} + \sum_{j \in M} \sum_{\substack{t \in L_j \\ t > m_I}} \Delta_{j,i_t^*} \mathbb{1}\{t \in I(j)\} \tag{38}$$

$$= \quad \sum_{j \in M_I} \Delta_{j,i_j^*} + \sum_{j \in M} \sum_{z=1}^{E_j} \Delta_{j,z} \sum_{\substack{t \in L_j^z \\ t > m_I}} \mathbb{1}\{t \in I(j)\} \tag{39}$$

Let us call

$$T_j^z(l_j^z) = \sum_{\substack{t \in L_j^z \\ t > m_I}} \mathbb{1}\{t \in I(j)\}$$

the total number of times we choose arm j in epoch z during the game (after initialization). Then, by taking the expectation of (39) we get

$$\mathbb{E}[R_n] = \sum_{j \in M_I} \Delta_{j,i_j^*} + \sum_{j \in M} \sum_{z=1}^{E_j} \Delta_{j,z} \, \mathbb{E}\left[T_j^z(l_j^z)\right]. \tag{40}$$

Therefore, finding an upper bound for the expected value of (38) can be accomplished by bounding the expected value of $T_j^z(l_j^z)$.

Second step: Decomposition of $T_j^z(l_j^z)$.

Recall that with $T_j(t-1)$ we indicate the number of times we played arm j before turn t starts. For any integer $u_{j,z}$, we can write

$$\begin{aligned} T_j^z(l_j^z) &= u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} \mathbb{1}\{t \in I(j), T_j(t-1) \geq u_{j,z}\} \\ &= u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} \mathbb{1}\left\{\widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} > \widehat{X}_{i_t^*} + \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}}, T_j(t-1) \geq u_{j,z}\right\} \\ &\leq u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} \sum_{k_j=u_{j,z}}^{t-s_j} \sum_{k_{i_t^*}=1}^{t-s_{i_t^*}} \mathbb{1}\left\{\widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{k_j}} > \widehat{X}_{i_t^*} + \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{k_{i_t^*}}}\right\}. \end{aligned}$$

Therefore we can find an upper bound for the expectation of $T_j^z(l_j^z)$ by finding an upper bound for the probability of the event

$$A = \left\{\widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{k_j}} > \widehat{X}_{i_t^*} + \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{k_{i_t^*}}}\right\}.$$

Third step: Upper bound for $\mathbb{E}[T_j^z(l_j^z)]$.

Using Proposition 1 and Proposition 2 we have that, by choosing $u_{j,z} = \max_{t\in\{s_j^z,\cdots,l_j^z\}} \left\lceil \frac{8\psi(j,t)\log(t-s_j^z)}{\Delta_{j,z}^2}\right\rceil$,

$$A \subset \left(\left\{\widehat{X}_{i_t^*} < \mu_{i_t^*} - \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{k_{i_t^*}}}\right\} \cup \left\{\widehat{X}_j > \mu_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{k_j}}\right\}\right). \tag{41}$$

Using Hoeffding's[4] bound we have that

$$\mathbb{P}\left(\widehat{X}_{i_t^*} < \mu_{i_t^*} - \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}}\right) \leq \exp\left\{-\frac{2k_{i_t^*}^2\psi(i_t^*,t)^2\frac{2\log(t-s_{i_t^*})}{k_{i_t^*}}}{k_{i_t^*}r^2}\right\} = (t-s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)}$$

$$\mathbb{P}\left(\widehat{X}_j > \mu_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}}\right) \leq \exp\left\{-\frac{2k_j^2\psi(j,t)^2\frac{2\log(t-s_j)}{k_j}}{k_j r^2}\right\} = (t-s_j)^{-\frac{4}{r^2}\psi(j,t)}.$$

Using the inclusion in (41) in combination with Hoeffding's bounds, we have that

$$\begin{aligned} \mathbb{E}\left[T_j^z(l_j^z)\right] &\leq u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} \sum_{k_j=u_{j,z}}^{l_j} \sum_{k_{i_t^*}=1}^{t-s_{i_t^*}} \mathbb{P}\left\{\widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{k_j}} > \widehat{X}_{i_t^*} + \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{k_{i_t^*}}}\right\} \\ &\leq u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} \sum_{k_j=u_{j,z}}^{t-s_j} \sum_{k_{i_t^*}=1}^{t-s_{i_t^*}} \left[(t-s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)} + (t-s_j)^{-\frac{4}{r^2}\psi(j,t)}\right] \\ &= u_{j,z} + \sum_{\substack{t\in L_j^z \\ t>m_I}} (t-s_{i_t^*})(t-s_j-u_{j,z}+1)\left[(t-s_j)^{-\frac{4}{r^2}\psi(j,t)} + (t-s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)}\right]. \end{aligned} \tag{42}$$

[4] **Hoeffding's bound:** Let $X_1,\cdots,X_n$ be r.v. bounded in $[a_i,b_i]$ $\forall i$. Let $\widehat{X} = \frac{1}{n}\sum_{i=1}^n X_i$ and $\mu = \mathbb{E}[\widehat{X}]$. Then, $\mathbb{P}\left(\widehat{X} - \mu \geq \varepsilon\right) \leq \exp\left\{-\frac{2n^2\varepsilon^2}{\sum_{i=1}^n (b_i-a_i)^2}\right\}$.

In our case, n is k_j or $k_{i_t^*}$, $b_i - a_i$ is r, μ is μ_j or $\mu_{i_t^*}$, and ε is $\psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}}$ or $\psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}}$.

Of course, we also have that the expected number of times the algorithm chooses arm j during epoch L_j^z is also bounded by the length of the epoch itself $l_j^z - s_j^z$ (this bound is useful in case the epoch is very short). Combining this with (42) we have that

$$\mathbb{E}\left[T_j^z(l_j^z)\right] \leq \min\left(l_j^z - s_j^z \ , \ u_{j,z} + \sum_{\substack{t \in L_j^z \\ t > m_I}} (t - s_{i_t^*})(t - s_j - u_{j,z} + 1)\left[(t-s_j)^{-\frac{4}{r^2}\psi(j,t)} + (t - s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)}\right] \right). \quad (43)$$

Fourth step: Get upper bound for $\mathbb{E}[R_n]$.

Combining (43) with (40) we get that the bound on the cumulative regret is given by

$$\begin{aligned}\mathbb{E}[R_n] \quad &\leq \quad \sum_{j \in M_I} \Delta_{j,i_j^*} \\ &+ \quad \sum_{j \in M} \sum_{z=1}^{E_j} \Delta_{j,z} \min\left(l_j^z - s_j^z \ , \ u_{j,z} + \sum_{\substack{t \in L_j^z \\ t > m_I}} (t - s_{i_t^*})(t - s_j - u_{j,z} + 1)\left[(t-s_j)^{-\frac{4}{r^2}\psi(j,t)} + (t - s_{i_t^*})^{-\frac{4}{r^2}\psi(i_t^*,t)}\right] \right).\end{aligned}$$

Notice that if $\psi(j,t) = 1$, $s_j = 0$ and $l_j > n \ \forall j, t$, you can recover the bound of the standard UCB algorithm used in the stochastic case. (Note that you should use $P > 2$ instead of 2 when r is not 1 to create the UCB.)

The results in Proposition 1 and 2 are similar to arguments used in Auer et al. [2002] for the proof of the regret bound for the UCB algorithm (here we have additional weighting of the upper confidence bound).

Proposition 1. *The event*

$$A = \left\{ \widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} > \widehat{X}_{i_t^*} + \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}} \right\}$$

is included in $B \cup C \cup D$, where

$$\begin{aligned} B &= \left\{ \widehat{X}_{i_t^*} < \mu_{i_t^*} - \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}} \right\} \\ C &= \left\{ \widehat{X}_j > \mu_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} \right\} \\ D &= \left\{ \mu_{i_t^*} - \mu_j < 2\psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} \right\} \end{aligned}$$

The inclusion $A \subset (B \cup C \cup D)$ intuitively means that if the algorithm is choosing to play suboptimal arm j at turn t, then it is underestimating the best arm available (event B), or it is overestimating arm j (event C), or it has not pulled enough times arm j to distinguish its performance from the one of arm i_t^* (event D).

For the sake of contradiction let us assume there exists $\omega \in A$ such that $\omega \in (B \cup C \cup D)^{\mathcal{C}}$. Then, for that ω, none of the inequalities that define the events B, C, and D would hold, i.e. (using, in order, the inequality in B, then the one in D, then the one in C):

$$\begin{aligned} \widehat{X}_{i_t^*} &\geq \mu_{i_t^*} - \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}} \\ &\geq \mu_j + 2\psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} - \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}} \\ &\geq \widehat{X}_j + \psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} - \psi(i_t^*,t)\sqrt{\frac{2\log(t-s_{i_t^*})}{T_{i_t^*}(t-1)}}, \end{aligned}$$

which contradicts $\omega \in A$.

The result in Proposition 2 is similar to the one used in Auer et al. [2002] for the proof of the regret bound for the UCB algorithm.

Proposition 2. *When*

$$T_j(t-1) \geq \left\lceil \frac{8\psi(j,t)\log(t-s_j)}{\Delta_{j,i_t^*}^2} \right\rceil$$

event D in Preposition 1 can not happen.

In fact,

$$
\begin{aligned}
& \mu_{i_t^*} - \mu_j - 2\psi(j,t)\sqrt{\frac{2\log(t-s_j)}{T_j(t-1)}} \\
\geq \quad & \mu_{i_t^*} - \mu_j - 2\psi(j,t)\sqrt{\frac{2\log(t-s_j)}{\left\lceil \frac{8\psi(j,t)\log(t-s_j)}{\Delta_{j,i_t^*}^2} \right\rceil}} \\
\geq \quad & \mu_{i_t^*} - \mu_j - 2\psi(j,t)\sqrt{\frac{\log(t-s_j)\Delta_{j,i_t^*}^2}{4\psi(j,t)\log(t-s_j)}} \\
= \quad & \mu_{i_t^*} - \mu_j - \Delta_{j,i_t^*} = 0.
\end{aligned}
$$

C Useful results

The result in Proposition 3 is similar to the one used in Auer et al. [2002] for the proof of the regret bound for the ε-greedy algorithm.

Proposition 3. *Let $\mu_i > \mu_j$ and let us define the following events:*

$$\begin{aligned} A &= \left\{\widehat{X}_j > \widehat{X}_i\right\}, \\ B &= \left\{\widehat{X}_i < \mu_i - \frac{\Delta(i,j)}{2}\right\}, \\ C &= \left\{\widehat{X}_j > \mu_j + \frac{\Delta(i,j)}{2}\right\}. \end{aligned}$$

Then,

$$A \subset (B \cup C)\,. \tag{44}$$

Intuitively, the inclusion in (44) means that we play arm j when we underestimate the mean reward of the best arm, or when we overestimate that of arm j. Assume for the sake of contradiction that there exists an element $\omega \in A$ that does not belong to $B \cup C$. Then, we have that $\omega \in (B \cup C)^C$

$$\Rightarrow \quad \omega \in \left(\left\{\widehat{X}_i < \mu_i - \frac{\Delta(i,j)}{2}\right\} \cup \left\{\widehat{X}_j > \mu_j + \frac{\Delta(i,j)}{2}\right\}\right)^C \tag{45}$$

$$\Rightarrow \quad \omega \in \left\{\widehat{X}_i \geq \mu_i - \frac{\Delta(i,j)}{2}\right\} \cap \left\{\widehat{X}_j \leq \mu_j + \frac{\Delta(i,j)}{2}\right\}. \tag{46}$$

By definition we have $\mu_i - \frac{\Delta(i,j)}{2} = \mu_i - \frac{\mu_i - \mu_j}{2} = \frac{\mu_i + \mu_j}{2} = \mu_j + \frac{\Delta(i,j)}{2}$. From the inequalities given in (46) it follows that

$$\widehat{X}_i \geq \mu_i - \frac{\Delta(i,j)}{2} = \mu_j + \frac{\Delta(i,j)}{2} \geq \widehat{X}_j,$$

but this contradicts our assumption that $\omega \in A = \left\{\widehat{X}_j > \widehat{X}_i\right\}$.
Therefore, all elements of A belong to $B \cup C$.

D Numerical results

D.1 Dataset

The dataset can be found on the Yahoo Webscope program. It contains files recording 15 days of article recommendation history. Each record shows information about the displayed article id, user features, timestamp and the candidate pool of available articles at that time. The displayed article id shows the arm that recommenders pick each turn. User features were not used, since our algorithms look for articles generally liked by everyone. Timestamp tells the time that event happens; along with the candidate pool of available articles, we can scan through the records and find out each article's lifespan.

D.2 Evaluation methodology

A unique property of this dataset is that the displayed article is chosen uniformly at random from the candidate article pool. Therefore, one can use an unbiased offline evaluation method Li et al. [2011] to compare bandit algorithms in a reliable way. However, in the initialization phase, we applied a simpler and faster method (Algorithm 4), since initialization only plays 25 turns in a game and we care more about what happens later on.

In order to apply these evaluation methods, after parsing the original text log into structured data frame, we made an event stream generator out of it. The event stream generator has a member method "next_event()" that gives us the next record in the data frame. The fields in the record give information about the event. For example, in the initialization phase we checked the "article" field of the records to see if that article had been played before.

Algorithm 4: Initialization

event stream $Stream$
number of turns as initialization m
$i \leftarrow 0$
while $i < m$ **do**
  $Record \leftarrow Stream.next\_event()$
  if $Record.article$ was not seen before **then**
    update expectation of $Record.article$
    $i \leftarrow i + 1$
  end if
end while

D.3 Parameter tuning

AG-L filters out a portion of articles that expire soon. This portion is a tunable parameter. We tested different values with a smaller size dataset and finally used 0.1 as the threshold. In UCB-L's upper confidence bound, $\psi(j,t) = clog(l_j - t + 1)$ and c is a tunable parameter. After tuning, we set $c = 0.011$ for later experiments.

D.4 UCB score function

The original expression for the modified upper confidence bound in UCB-L is $X + \psi\sqrt{\frac{2log(t-s)}{T}}$. In the experiment, we used $X + \psi\sqrt{\frac{2log(t-s+1)}{T}}$ because sometimes an article is chosen the turn it becomes available and $t = s$, leading to an invalid value.

D.5 Timestamp vs Turn number

The original dataset records each action and reward along its corresponding timestamp, which we pre-scanned to get the articles' lifespan. But our algorithms count articles' lifespan in "turns." In this offline evaluation setting, a lot of events are discarded if they don't match the actions our algorithms pick, and there is no direct relation between an event's timestamp and turn number.

For AG-L, each time we rank articles by their remaining life, and filter out those with shorter remaining life. Since "timestamp" and "turn number" are positively correlated, ranking by "timestamp" and ranking by "turn number" would

almost give the same result ("almost" because sometimes consecutive events have equal timestamps, but later events should happen later in the game and are associated with bigger turn numbers). Then we can simulate AG-L using the timestamps.

In AG-L, if we use the average lifespan of previous arms as the estimation for current arms, we will be comparing arms by their birth time – all current arms are expected to have the same life length and the later an arm is born, the shorter remaining life it has.

However, UCB-L needs the exact turn number to compute the modified upper confidence bound. We can record the turn number when an article appears, but there's no way to know at which turn this article expires – We only know at what timestamp it disappears, but we can't transfer timestamps to turn numbers. Therefore, we only simulated UCB-L using life estimation.

For a short period in the early game we didn't have the life estimation because we hadn't yet seen an expired article. Also, sometimes our estimated life length $\hat{L}$ is too small, then $\hat{L}+s-t+1 \leq 0$, leading $\psi(j,t) = clog(l-t+1) = clog(\hat{L}+s-t+1)$ to an invalid value. In these cases we took $l-t+1 = \hat{L}+s-t+1 = 0$ and used only X as the upper confidence bound.

D.6 Contextual algorithm

Algorithm 5 is a similar adaptation of the LinUCB algorithm introduced by Li et al. [2010] to the mortal setting. Also in this case, the function $\psi(j,t)$ regulates the amplitude of the upper confidence bound above the estimated mean according to the remaining life of the arm. As before, new arms are initialized by using the average performance of past arms (i.e., if in the past a lot of bad arms appeared, new arms are considered more likely to be bad, and vice-versa if lots of good arms appeared in the past).

Algorithm 5: LinUCB-L algorithm

Input : number of rounds n, initial set of arms M_I, set M_t of available arms at time, rewards range $[a,b]$, dimension d (context space dimension + arms space dimension)

Initialization: For each $j \in M_I$, $A_j = I_d$, $b_j = 0_{d\times 1}$

for $t = 1$ **to** n **do**

  Get context x_t (or $x_{t,j}$ if each arm gets its context);

  for $j = 1$ **to** m_t **do**

    Set $\hat{\theta}_j = A_j^{-1} b_j$;

    Set $UCB_j = \hat{\theta}_j^T x_t + \psi(j,t)\sqrt{x_t^T A_j^{-1} x_t}$;

  end

  Play arm $j = \text{argmax}_i\, UCB_i$;

  Get reward $X_j(t)$;

  Update $A_j = A_j^{-1} + x_t x_t^T$;

  Update $b_j = b_j + r_t x_t$;

end

We have noticed that the contextual algorithm was not useful for the features made available in the Yahoo! Webscope Dataset, so for the experiments we used the non-contextual version presented in the main paper.

E Notation summary

- M_t as the set of all available arms at turn t;
- M_I the set of arms that are initialized;
- m_t: number of arms available at time t;
- n: total number of rounds;
- $X_j(t)$: random reward for playing arm j at time t;
- μ_*: mean reward of the optimal arm ($\mu_* = \max_{1 \leq j \leq m} \mu_j$);
- $\Delta(i, j)$: difference between the mean reward of arm i and arm j ($\Delta(i, j) = \mu_i - \mu_j$);
- $\hat{X}_j$: current estimate of μ_j;
- I_j: set of turns when arm j is played;
- κ_j turn at which arm j is initialized;
- $T_j(t-1)$: r.v. of the number of times arm j has been played before round t starts;
- $\mathcal{H}_{t-1}$: set of all possible histories h (after deterministic initialization) of the game up to turn $t-1$;
- $U_s(h, i_s)$: upper bound on the probability that arm i_s was pulled at time s given the history of pulls h up to time $s-1$;
- $u_s(h, i_s)$: upper bound on the probability that arm i_s is considered to be the best arm at time s given the history of pulls h up to time $s-1$;
- $f_{M(h,s)}(g(p))$: the PDF (or PMF) of the maximum of the estimated mean rewards at time s given that each arm has been pulled according to history h up to time $s-1$;
- $g(p)$: linear transformation $g(p) = b + (a-b)p$;
- $U_t(h, j)$: upper bound on the probability that arm j was pulled at time t given the history of pulls h up to time $t-1$;
- $u_t(h, j)$: upper bound on the probability that arm j is considered to be the best arm at time t given the history of pulls h up to time $t-1$;
- R_n: total regret at round n.