
A Relative Exponential Weighing Algorithm for Adversarial Utility-based Dueling Bandits

Pratik Gajane
Tanguy Urvoy
Fabrice Clérot

Orange-labs, Lannion, France

PRATIK.GAJANE@ORANGE.COM
TANGUY.URVOY@ORANGE.COM
FABRICE.CLEROT@ORANGE.COM

Abstract

We study the K -armed dueling bandit problem which is a variation of the classical Multi-Armed Bandit (MAB) problem in which the learner receives only relative feedback about the selected pairs of arms. We propose an efficient algorithm called *Relative Exponential-weight algorithm for Exploration and Exploitation* (REX3) to handle the adversarial utility-based formulation of this problem. We prove a finite time expected regret upper bound of order $\mathcal{O}(\sqrt{K \ln(K)T})$ for this algorithm and a general lower bound of order $\Omega(\sqrt{KT})$. At the end, we provide experimental results using real data from information retrieval applications.

1. Introduction

The K -armed dueling bandit problem is a variation of the classical Multi-Armed Bandit (MAB) problem introduced by Yue and Joachims (2009) to formalize the exploration/exploitation dilemma in learning from preference feedback. In its utility-based formulation, at each time period, the environment sets a bounded value for each of K arms. Simultaneously the learner selects two arms and wagers that one of the arms will be *better* than the other. The learner only sees the outcome of the *duel* between the selected arms (i.e. the feedback indicates which of the selected arms has better value) and receives the average of the rewards of the selected arms. The goal of the learner is to maximize her *cumulative reward*. The difficulty of this problem stems from the fact that the learning algorithm has no way of directly observing the reward of its actions. This is a perfect example of *partial monitoring problem* as defined in Piccolboni and Schindelhauer (2001).

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Relative feedback is naturally suited to many practical applications like user-perceived product preferences, where a relative perception: “*A is better than B*” is easier to obtain than its absolute counterpart: “*A value is 42, B is worth 33*”. Another important application of dueling bandits comes from information retrieval systems where users provide *implicit feedback* about the provided results. This implicit feedback is collected in various ways e.g. a click on a link, a tap, or any monitored action of the user. In all these ways however, this kind of feedback is often strongly biased by the model itself (the user cannot click on a link which was not proposed).

To remove this bias in search engines, Radlinski and Joachims (2007) propose to interleave the outputs of different ranking models: the model which scores a click wins the duel. The accuracy of this *interleave filtering method* was highlighted in several experimental studies (Radlinski and Joachims, 2007; Joachims et al., 2007; Chapelle et al., 2012).

Main contribution

The main contribution of this article is an algorithm designed for the adversarial utility-based dueling bandit problem in contrast to most of the existing algorithms which assume a stochastic environment.

Our algorithm, called *Relative Exponential-weight algorithm for Exploration and Exploitation* (REX3), is a non-trivial extension of the *Exponential-weight algorithm for Exploration and Exploitation* (EXP3) algorithm (Auer et al., 2002b) to the dueling bandit problem. We prove a finite time expected regret upper bound of order $\mathcal{O}(\sqrt{K \ln(K)T})$ and develop an argument initially proposed by Ailon et al. (2014) to exhibit a general lower bound of order $\Omega(\sqrt{KT})$ for this problem.

These two bounds correspond to the original bounds of the classical EXP3 algorithm and the upper bound strictly improves from the $\tilde{\mathcal{O}}(K\sqrt{T})$ obtained by existing generic

partial monitoring algorithms.¹

Our experiments on information retrieval datasets show that the anytime version of REX3 is a highly competitive algorithm for dueling bandits, especially in the initial phases of the runs where it clearly outperforms the state of the art.

Outline

This article is organized as follows: In section 2, we give a brief survey on dueling bandits with section 2.3 dedicated to the adversarial case. Most notations and formalizations are introduced in this section. Secondly, in section 3, we formally describe REX3 with its pseudo-code. Then, in section 4.1, we provide the upper bound on the expected regret of REX3. Furthermore, in section 4.2, we provide the lower bound on the regret of any algorithm attempting to solve the adversarial utility-based dueling bandit problem. Section 5 begins with an empirical study of the bound given in section 4.1. It then provides comparisons of REX3 with state-of-the-art algorithms on information retrieval datasets. The conclusion is provided in section 6.

2. Previous Work and Notations

The conventional MAB problem has been well studied in the stochastic setting as well as the (oblivious) adversarial setting (see Cesa-Bianchi and Lugosi, 2006; Bubeck and Cesa-Bianchi, 2012). These MAB algorithms are designed to optimize exploration and exploitation in order to control the *cumulative regret* which is the difference between the gain of a reference strategy and the actual gain of the algorithm.

2.1. Exponential-weight algorithm for Exploration and Exploitation

Of particular interest are the *Exponential-weight algorithm for Exploration and Exploitation* (EXP3) and its variants presented by Auer et al. (2002b) for the adversarial bandit setting. For a fixed horizon T and K arms, the EXP3 algorithm provides an expected cumulative regret bound of order $\mathcal{O}(\sqrt{K \ln(K)T})$ against the best single-arm strategy. This algorithm is indeed *adversarial* because it does not require a stochastic assumption on the rewards. It is although not *anytime* because it requires the knowledge of the horizon T to run properly. A “*doubling trick*” solution is proposed by Auer et al. (2002b) to preserve the regret bound when T is unknown. It consists of running EXP3 in a carefully designed sequence of increasing *epochs*. Another elegant solution was later proposed by Seldin et al. (2012) for the same purpose.

¹The notation $\tilde{\mathcal{O}}(\cdot)$ hides logarithmic factors.

2.2. Previous work on stochastic dueling bandits

The dueling bandits problem is recent, although related to previous works on computing with noisy comparison (see for instance Karp and Kleinberg, 2007). This problem also falls under the framework of *preference learning* (Freund et al., 2003; Liu, 2009; Fürnkranz and Hüllermeier, 2010) which deals with learning of (predictive) preference models from observed (or extracted) preference information i.e. relative feedback which specifies which of the chosen alternatives is preferred. Most of the articles hitherto published on dueling bandits consider the problem under a stochastic assumption.

Yue and Joachims (2009) propose an algorithm called *Dueling Bandit Gradient Descent* (DBGD) to solve a version of the dueling bandits problem where context information is provided. They approach (*contextual*) dueling bandits as an on-line convex optimization problem.

Yue et al. (2012) propose an algorithm called *Interleaved Flitering* (IF). Their formulation is stochastic and *matrix-based*: for each pair (i, j) of arms, there is an unknown probability $P_{i,j}$ for i to win against j . This *preference matrix* P of size $K \times K$ must satisfy the following symmetry property:

$$\forall i, j \in \{1, \dots, K\}, \quad P_{i,j} + P_{j,i} = 1 \quad (1)$$

Hence on the diagonal: $P_{i,i} = \frac{1}{2} \quad \forall i \in \{1, \dots, K\}$. Let i^* be the “best arm” (as we will see later, this best arm coincides with the notion of *Condorcet winner*). Yue et al. (2012) define the regret incurred at the time instant t when arms a and b are pulled as:

$$r'_{a,b} = \frac{P_{i^*,a} + P_{i^*,b} - 1}{2} \in (0, \frac{1}{2}) \quad (2)$$

We will call this regret a *Condorcet regret*.

For the IF algorithm to work, the P matrix is expected to satisfy several strong assumptions: *strict linear ordering*, *stochastic transitivity*, and *stochastic triangular inequality*. Under these three assumptions IF is guaranteed to suffer an expected cumulative regret of order $\mathcal{O}(K \log T)$.

Yue and Joachims (2011) introduce *Beat The Mean* (BTM), an algorithm which proceeds by successive elimination of arms. This algorithm is less constrained than IF as it also applies to a relaxed setting where the preference matrix can slightly violate the stochastic transitivity assumption. Its cumulative regret bound is of order $\mathcal{O}(\gamma^T K \log T)$ where γ is here a known parameter.

Urvoy et al. (2013) propose a generic algorithm called SAVAGE (for Sensitivity Analysis of VArables for Generic Exploration) which does away with several assumptions made in the previous algorithms e.g. existence of inherent values of arms, existence of a linear order amongst arms. In

this general setting, the SAVAGE algorithm obtains a regret bound of order $\mathcal{O}(K^2 \log T)$. The key notions they introduce for dueling bandits are the *Copeland*, *Borda* and *Condorcet* scores (Charon and Hudry, 2010). The Borda score of an arm i on a preference matrix P is $\sum_{j=1}^K P_{i,j}$ and its Copeland score is $\sum_{j=1}^K \mathbb{I}[P_{i,j} > \frac{1}{2}]$ (We use $\mathbb{I}[\dots]$ to denote the indicator function). If an arm has a Copeland score of $K - 1$, which means that it defeats all the other arms in the long run, it is called a *Condorcet winner*. The existence of a Condorcet winner is the minimum assumption required for the *Condorcet regret* as defined on equation (2) to be applicable. There exists however some datasets like MSLR30K (2012) where this Condorcet condition is not satisfied.

Zoghi et al. (2014a) extend the *Upper Confidence Bound* (UCB) algorithm (Auer et al., 2002a) and propose an algorithm called *Relative Upper Confidence Bound* (RUCB) provided that the preference matrix admits a Condorcet winner. They retrieve an $\mathcal{O}(K \log T)$ bound under this sole assumption. Unlike the previous algorithms, RUCB is an anytime dueling bandits algorithm since it does not require the time horizon T as input.

Ailon et al. (2014) propose three methods (DOUBLER, MULTISMB, and SPARRING) to reduce the stochastic *utility-based dueling bandits* problem to the conventional MAB problem. A stochastic dueling bandits problem is *utility-based* if the preference is the result of comparisons of the individual utility/reward of the arms. This is a strong restriction from the general – *matrix-based* – formulation of the problem.

More formally, there are K probability distributions v_1, \dots, v_K over $[0, 1]$ associated respectively with arms $1, \dots, K$. Let μ_1, \dots, μ_K be the respective means of v_1, \dots, v_K . When an arm a is pulled, its reward/utility x_a is drawn from the corresponding distribution v_a ². Let $i^* \in \arg \max \mu_i$ be an optimal arm. The regret incurred at the time instant t when arms a and b are pulled is defined as:

$$r_{a,b}(t) = \frac{2x_{i^*} - x_a - x_b}{2} \quad (3)$$

We will call this regret a *bandit regret*. With randomized tie-breaking we can rebuild the preference matrix:

$$P_{i,j} = \mathbb{P}(x_i > x_j) + \frac{1}{2} \mathbb{P}(x_i = x_j)$$

When all v_i are Bernoulli laws, this reduces to:

$$P_{i,j} = \frac{\mu_i - \mu_j + 1}{2} \quad (4)$$

²Note that we frequently drop the time index when it is unnecessary or clear from context. For instance we simply write $x_a(t)$ or simply x_a instead of $x_{a_t}(t)$.

Note that if $\mu_i > \mu_j$ for some arms i and j , then $P_{i,j} > \frac{1}{2}$. The best arm in the usual bandit sense hence coincides with the Condorcet winner (which turns out to be the Borda winner too) on the matrix formulation and the expected bandit regret is twice the Condorcet regret as defined in (2):

$$\mathbb{E}r_{a,b} = \frac{2\mu_{i^*} - \mu_a - \mu_b}{2} = P_{i^*,a} + P_{i^*,b} - 1 = 2r'_{a,b}$$

Several other models and algorithms have been proposed since to handle stochastic dueling bandits. We can cite (Busa-Fekete et al., 2013; 2014; Zoghi et al., 2014b; 2015). See also (Busa-Fekete and Hüllermeier, 2014) for an extensive survey of this domain.

2.3. Adversarial dueling bandits

The bibliography on stochastic dueling bandits is flourishing, but the results about adversarial dueling bandits remain quite scarce.

A utility-based formulation of the problem is however proposed in Ailon et al. (2014, section 6). In this setting, as in classical adversarial MAB, the environment chooses before-hand an horizon T and a sequence of utility/reward vectors $\mathbf{x}(t) = (x_1(t), \dots, x_K(t)) \in [0, 1]^K$ for $t = 1, \dots, T$. The learning algorithm aims at controlling the bandit regret against the best single-arm strategy, as defined in (3), by choosing properly the pairs of arms (i, j) to be compared.

To tackle this problem, Ailon et al. (2014) suggest to apply the SPARRING reduction algorithm, although originally designed for stochastic settings, with an adversarial bandit algorithm like EXP3 as a black-box MAB. According to the authors, the SPARRING reduction preserves the $\mathcal{O}(\sqrt{KT \ln K})$ upper bound of EXP3. This algorithm uses two separate MABs (one for each arm). As a consequence, when it gets a relative feedback about a duel (i, j) , the left instantiation of EXP3 only updates its weight for arm i while the right instantiation only updates for j . The algorithm we propose improves from this solution by centralizing information for both arms on a single weight vector.

As mentioned earlier, the dueling bandits problem is a special instance of a partial monitoring game (Piccolboni and Schindelhauer, 2001; Cesa-Bianchi and Lugosi, 2009; Bartók et al., 2014). A partial monitoring game is defined by two matrices: a loss matrix \mathcal{L} and a feedback matrix \mathcal{H} . These two matrices are known by the learner. At each round, the learner chooses an action a while the environment simultaneously chooses an outcome (say \mathbf{x}). The learner receives a feedback $\mathcal{H}(a, \mathbf{x})$ and suffers (in a blind manner) a loss $\mathcal{L}(a, \mathbf{x})$.

It is straightforward to encode the classical MAB as a finite partial monitoring game: the actions are arms indices $a \in \{1, \dots, K\}$ while the environment outcomes

are reward vectors $\mathbf{x}(t) = (x_1(t), \dots, x_K(t)) \in [0, 1]^K$. The loss and feedback matrices are respectively defined by $\mathcal{L}(a, \mathbf{x}) = -x_a$ and $\mathcal{H}(a, \mathbf{x}) = x_a$. For the utility-based dueling bandits problem, the learner's actions are the duels $(a, b) \in \{1, \dots, K\}^2$ and the environment outcomes are reward vectors. If we constrain the rewards to be binary it turns out to be a finite partial monitoring game. The Loss matrix is defined by $\mathcal{L}((a, b), \mathbf{x}) = -(x_a + x_b)/2$ and the feedback by $\mathcal{H}((a, b), \mathbf{x}) = \psi(x_a - x_b)$ where ψ is a non-decreasing transfer function such that $\psi(0) = 0$ (usually $\psi(x) = \llbracket x > 0 \rrbracket$ or $\psi(x) = x$).

Several generic partial monitoring algorithms were recently proposed for both stochastic and adversarial settings (see Bartók et al., 2014, for details). If we except GLOBAL-EXP3 (Bartók, 2013) which tries to capture more finely the structure of the games, these algorithms only focus on the time bound and perform inefficiently when the number of actions grows.

In a dueling bandit the number of non-duplicate actions is actually $K(K+1)/2$ and these algorithms, including GLOBALEXP3, provide at best a $\tilde{\mathcal{O}}(K\sqrt{T})$ regret guarantee. The dedicated algorithm that we propose is using the preference feedback more efficiently.

3. Relative Exponential-weight Algorithm for Exploration and Exploitation

The pseudo-code for the algorithm we propose is given in Algorithm 1. As previously stated on Section 2.3, this algorithm is designed to apply for the adversarial utility-based dueling bandits problem.

It is similar to the original EXP3 from step 1 to step 6 where it computes a distribution $\mathbf{p}(t) = (p_1(t), \dots, p_K(t))$ which is a mixture of a normalized weighing of the arms $w_i / \sum_i w_i$ and a uniform distribution $1/K$. As in EXP3, this uniform probability is introduced to ensure a minimum exploration of all arms.

At step 7, the algorithm draws two arms a and b independently according to $\mathbf{p}(t)$. At step 8, the algorithm gets $\psi(x_a - x_b)$ as relative feedback. Note that, since arms are drawn with replacement, we may have $a = b$, in which case the algorithm will get no information. This event is indeed expected to become frequent when the $\mathbf{p}(t)$ distribution becomes peaked around the best arms. This necessity for a regret-minimizing dueling bandits algorithm to renounce getting information when confident about its decision is a structural bias toward exploitation that is not encountered in classical bandits.

Step 8 is the big difference from EXP3; because we only have access to the relative $\psi(x_a - x_b)$ value, we have no mean to estimate the individual rewards x_a or x_b . There

is however a solution to circumvent this problem: the best arm in expectation at time t is not only the one which maximizes the absolute reward. It is indeed the one which maximizes the regret of any fixed strategy $\pi(t)$ against it:

$$\arg \max_i x_i(t) = \arg \max_i (x_i(t) - \mathbb{E}_{a \sim \pi(t)} x_a).$$

This reference strategy could be a single-arm or uniform strategy but playing a suboptimal strategy to get a reference has a cost in terms of regret. One of our contributions is to show that the algorithm may use its own strategy as a reference.

At step 9, the condition $a \neq b$ is only a slight improvement for matrix-based dueling bandits where the outcome of a duel of an arm against itself is randomized as in (4).

At steps 10 and 11, the weights of the played arms are updated. This update process is the core of our algorithm, it will be detailed in Section 4.

Step 13 is only required for the anytime version of the algorithm. It will be explained in section 5.2.

Algorithm 1 REX3: Exp3 with relative feedback

- 1: **Parameters:** $\gamma \in (0, 1]$
 - 2: **Initialization:** $w_i(1) = 1$ for $i = 1, \dots, K$.
 - 3: **for** $t = 1, 2, \dots$ **do**
 - 4: **for** $i = 1, \dots, K$ **do**
 - 5: Set $p_i(t) \leftarrow (1 - \gamma) \frac{w_i(t)}{\sum_{j=1}^K w_j(t)} + \frac{\gamma}{K}$
 - 6: **end for**
 - 7: Pull two arms a and b chosen independently according to the distribution $(p_1(t), \dots, p_K(t))$.
 - 8: Receive relative feedback $\psi(x_a - x_b) \in [-1, +1]$
 - 9: **if** $a \neq b$ **then**
 - 10: Set $w_a(t+1) \leftarrow w_a(t) \cdot e^{\frac{\gamma}{K} \frac{\psi(x_a - x_b)}{2p_a}}$
 - 11: Set $w_b(t+1) \leftarrow w_b(t) \cdot e^{-\frac{\gamma}{K} \frac{\psi(x_a - x_b)}{2p_b}}$
 - 12: **end if**
 - 13: Update γ (for anytime version)
 - 14: **end for**
-

4. Analysis

For the analysis, we focus on the simple case where ψ is the identity. It provides a ternary *win/tie/loss* feedback if we assume binary rewards. The main difference between EXP3 and our algorithm is at steps 10 and 11 of Algorithm 1, where we update the weights according to the duel outcome: the winning arm is gratified while the loser is penalized. This ‘punitive’ approach of exponential weighing departs from EXP3 and other weighing algorithms which gratify the most rewarding arms while kindly ignoring the non-rewarding ones (Freund and Schapire, 1999; Cesa-Bianchi and Lugosi, 2006).

4.1. Upper bound for REX3

In this section, we provide a finite-horizon non-stochastic upper bound on the expected regret against the best single action policy.

The steps 10-11 on Algorithm 1 are equivalent to operating for each arm i an update of the form:

$$w_i(t+1) = w_i(t) \cdot e^{\frac{\gamma}{K} \hat{c}_i(t)}$$

where

$$\hat{c}_i(t) = \mathbb{I}[i = a] \frac{\psi(x_a - x_b)}{2p_a} + \mathbb{I}[i = b] \frac{\psi(x_b - x_a)}{2p_b} \quad (5)$$

One big difference with EXP3 is that $\hat{c}_i(t)$ not an estimator of the reward $x_i(t)$. We instead have:

Lemma 1.

$$\mathbb{E}[\hat{c}_i(t) | (a_1, b_1), \dots, (a_{t-1}, b_{t-1})] = \mathbb{E}_{a \sim p(t)} \psi(x_i(t) - x_a(t))$$

If ψ is the identity then $\mathbb{E}\hat{c}_i(t) = x_i(t) - \mathbb{E}_{a \sim p(t)} x_a(t)$ in which case we estimate the expected instantaneous regret of the algorithm against arm i . If we rather take $\psi(x) = \mathbb{I}[x > 0]$, then $\mathbb{E}\hat{c}_i(t) = \mathbb{P}_{a \sim p(t)}(x_i(t) > x_a(t))$, i.e. the probability for the algorithm to select an arm defeated by i .

Let $\mathbb{G}_{max} = \max_i \sum_{t=1}^T x_i(t)$ be the best single-arm gain, and let $\mathbb{G}_{alg} = \frac{1}{2} \sum_{t=1}^T x_a(t) + x_b(t)$ be the gain of the algorithm. Let $\mathbb{E}\mathbb{G}_{unif} = \frac{1}{K} \sum_{t=1}^T \sum_{i=1}^K x_i(t)$ be the average value of the game (i.e. the expected gain of the uniform sampling strategy).

Theorem 1. *If the transfer function ψ is the identity and $\gamma \in (0, \frac{1}{2})$, then,*

$$\mathbb{G}_{max} - \mathbb{E}(\mathbb{G}_{alg}) \leq \frac{K}{\gamma} \ln(K) + \gamma\tau$$

where $\tau = e \cdot \mathbb{E}\mathbb{G}_{alg} - (4-e) \cdot \mathbb{E}\mathbb{G}_{unif}$.

We give a sketch of proof for this result in section A.

Provided that $\mathbb{E}\mathbb{G}_{alg} \leq \mathbb{G}_{max}$ and $\mathbb{E}\mathbb{G}_{unif} \geq \mathbb{G}_{min}$, where $\mathbb{G}_{min} = \min_i \sum_{t=1}^T x_i(t)$ is the gain of the worst single-arm strategy, we can simplify the bound into:

Corollary 1.

$$\mathbb{G}_{max} - \mathbb{E}\mathbb{G}_{alg} \leq \frac{K \ln K}{\gamma} + \gamma(e\mathbb{G}_{max} - (4-e)\mathbb{G}_{min})$$

As in (Auer et al., 2002b, section 3), since $\frac{K}{\gamma} \ln(K) + \gamma\tau$ is convex, we can obtain the optimal γ on $(0, \frac{1}{2})$:

$$\gamma^* = \min \left\{ \frac{1}{2}, \sqrt{\frac{K \ln(K)}{\tau}} \right\} \quad (6)$$

Substituting γ in Corollary 1 with its optimal value from eq. (6) we obtain:

$$\mathbb{G}_{max} - \mathbb{E}(\mathbb{G}_{alg}) \leq 2\sqrt{K \ln(K)} [e\mathbb{G}_{max} - (4-e)\mathbb{G}_{min}]$$

Hence:

Corollary 2. *When $\gamma = \min \left\{ \frac{1}{2}, \sqrt{\frac{K \ln(K)}{\tau}} \right\}$, the expected regret of REX3 (Algorithm 1) is $\mathcal{O} \left(\sqrt{K \ln(K) T} \right)$.*

The upper bound of REX3 for adversarial utility-based dueling bandits is hence the same as the one of EXP3 for adversarial MABs. For a high-number of arms or a short term horizon, this bound is competitive against the $\mathcal{O}(K \ln(T))$ or $\mathcal{O}(K^2 \ln(T))$ existing bounds for stochastic dueling bandits.

4.2. Lower bound for dueling bandits algorithms

To provide a lower bound on the regret of any dueling bandits algorithm, we use a reduction to the classical MAB problem suggested by Ailon et al. (2014).

Algorithm 2 Reduction to classical MAB

```

1: DBA.init()
2: Set  $t = 1$ 
3: repeat
4:    $(a_t, b_{t+1}) \leftarrow$  DBA.decide()
5:    $x_{a_t} \leftarrow$  CBE.get_reward()
6:    $x_{b_{t+1}} \leftarrow$  CBE.get_reward()
7:   DBA.update( $(a_t, b_{t+1}), (x_{a_t} - x_{b_{t+1}})$ )
8:    $t = t + 2$ 
9: until  $t \geq T$ 
    
```

Algorithm 2 gives an explicit formulation of this reduction by using a generic dueling bandits algorithm (DBA) as a black-box having the following public sub-routines: `init()`, `decide()` and `update()`. The classical bandit environment (CBE) provides `get_reward()` which returns the reward of the input arm. The expected classical-bandit gain of Algorithm 2 will be twice the expected gain of the black-box dueling bandits it uses.

It is important to note that this reduction only works for stochastic settings where the expected reward of each arm remains the same across time instants. According to Theorem 5.1 given by Auer et al. (2002b, section 5), for $K \geq 2$, the expected regret in the classical bandit setting is $\Omega(\sqrt{KT})$ (assuming T is large enough i.e. $T \geq \sqrt{KT}$). Since this result is obtained with a stationary stochastic distribution, by extension, the expected regret in the dueling bandits setting cannot be less than $\Omega(\sqrt{KT})$.

Theorem 2. *For any number of actions $K \geq 2$ and large enough time horizon T (i.e. $T \geq \sqrt{KT}$), there exists a*

distribution over assignments of rewards such that the expected cumulative regret of any utility-based dueling bandits algorithm cannot be less than $\Omega(\sqrt{KT})$.

5. Experiments

To evaluate REX3 and other dueling bandits algorithms, we have applied them to the online comparison of rankers for search engines by *interleaved filtering* (Radlinski and Joachims, 2007). A search-engine ranker is a function that orders a collection of documents according to their relevancy to a given user search query. By interleaving the output of two rankers and tracking on which ranker’s output the user did click, we are able to get an unbiased feedback about the relative quality of these two rankers. Given K rankers, the problem of finding the best ranker is indeed a K -armed dueling bandits.

In order to obtain reproducible and comparable results, we adopted the stochastic matrix-based experiment setup already employed by Yue and Joachims (2011); Zoghi et al. (2014a;b; 2015) with both the cumulative Condorcet regret as defined by Yue et al. (2012) and the *accuracy* i.e. the best arm hit-rate over the runs: $\frac{1}{N} \sum_n \mathbb{I}((a, b) = (i^*, i^*))$.

This experimental setup uses real search engines’ logs to build empirical preference matrices. The duel outcomes are then simulated on these matrices. We used several preference matrices issued from namely: ARXIV dataset (Yue and Joachims, 2011), LETOR NP2004 dataset (Liu et al., 2007), and MSLR30K dataset. The last dataset distinguishes three kinds of queries: informational, navigational and perfect-hit navigational (MSLR30K, 2012). These matrices are courtesy of Zoghi et al. (2014b)’s authors.

5.1. Empirical validation of Corollary 1

We have used LETOR NP2004 and MSLR30K datasets (restricted to 64 rankers) to compare the average Condorcet regret of 100 runs of REX3 with $T = 10^5$ to the corresponding halved³ theoretical bounds from Corollary 1 for various values of γ . The results of this experiment are summarized in Figure 1. We plotted two theoretical curves: one with a conservative $\mathbb{G}_{max} = T/2$, and a riskier one with $\mathbb{G}_{max} = T/4$. This experiment illustrates the dual impact of the γ parameter on the exploration/exploitation tradeoff: a low value reduces both the exploration and the reactivity of the algorithm to unexpected feedbacks and a high value tends to uniformize exploration while increasing reactivity. It also shows that the theoretical optimal γ^* we obtain with Equation (6) is a good guess even with a conservative upper-bound for \mathbb{G}_{max} .

³ As mentioned at the end of section 2.2, the utility-based bandit regret is indeed twice the Condorcet regret as defined in (2).

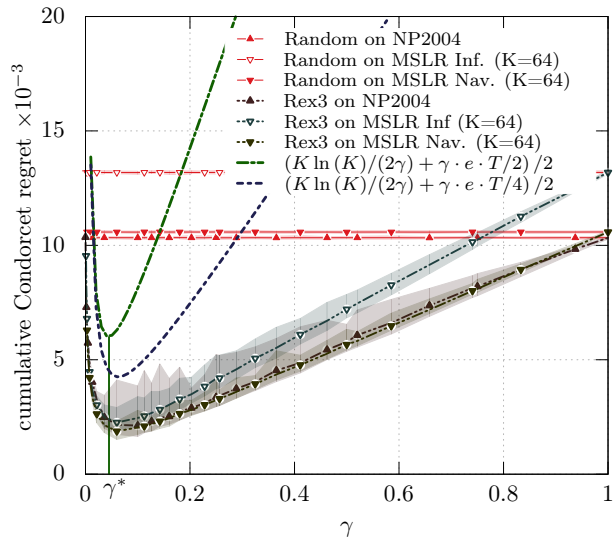


Figure 1. Empirical validation of Corollary 1. The colored areas around the curves show the minimal and maximal values over 100 runs.

5.2. Interleave filtering simulations

For our experiments we have considered the following state of the art algorithms: BTM (Yue and Joachims, 2011) with $\gamma = 1.1$ and $\delta = 1/T$ (explore-then-exploit setting), Condorcet-SAVAGE (Urvoy et al., 2013) with $\delta = 1/T$, RUCB (Zoghi et al., 2014a) with $\alpha = 0.51$, and SPARRING coupled with EXP3 (Ailon et al., 2014). We also took the uniform sampling strategy RANDOM as a baseline. We considered three versions of REX3: two non-anytime versions where the optimal γ^* is computed beforehand according to (6) with \mathbb{G}_{max} set respectively to $T/2$ and $T/10$ and one anytime version where γ^* is recomputed at each time step according to (6) (see Seldin et al., 2012, for details about this form of “doubling trick”).

A point which makes the comparison difficult is that some algorithms are anytime while others require the horizon as input. For anytime algorithms, namely RANDOM, RUCB and REX3 with adaptive γ , we displayed the average over 100 runs of the progressive accumulation of regret while for non-anytime algorithms, namely BTM, CSAVAGE, SPARRING and other versions of REX3, we displayed the average over 50 runs of the final cumulative regret for several fixed and known horizons. This protocol is slightly favorable to non-anytime algorithms which benefit from more information. However, for elimination algorithms like BTM and CSAVAGE the difference between the anytime regret and the non-anytime regret is small. For adversarial algorithms like SPARRING and REX3 the “doubling trick” can be applied to make them anytime: the adaptive γ version of REX3 is an example of such a fixed-to-anytime transformation.

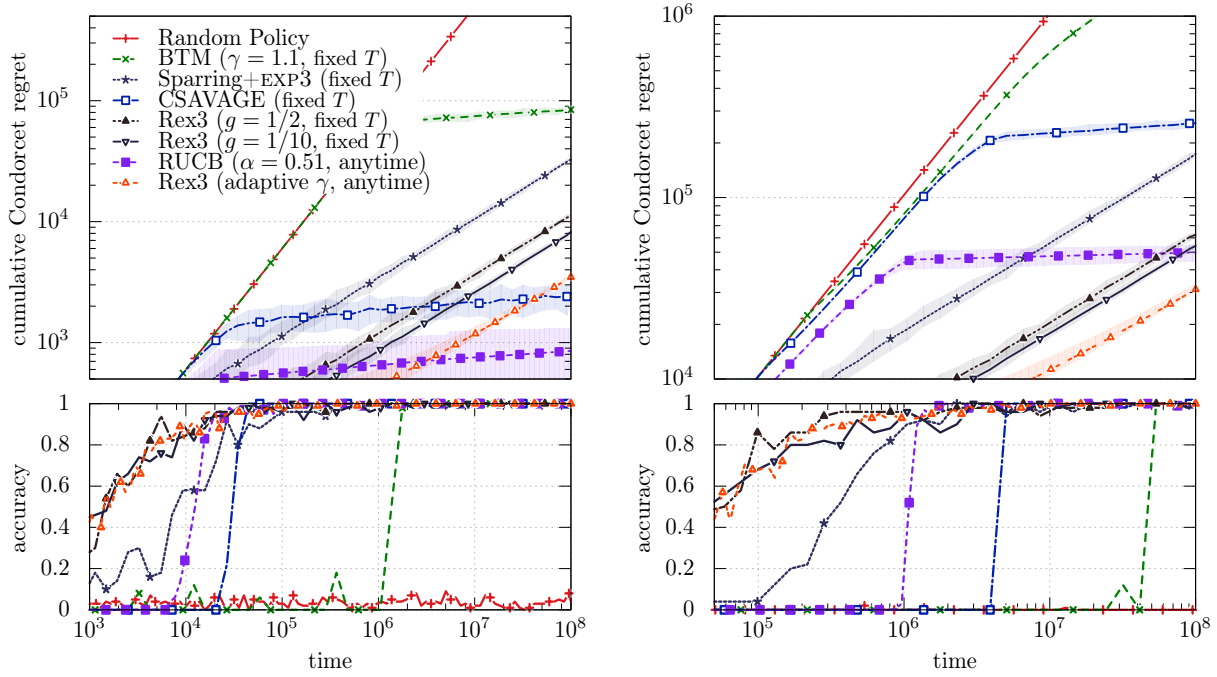


Figure 2. Regret and accuracy plots averaged over 100 runs (50 runs for fixed-horizon algorithms) respectively on ARXIV dataset (6 rankers) and LETOR NP2004 dataset (64 rankers). On regret plots, both time and regret scales are logarithmic (\sqrt{t} hence appears as $t/2$). The colored areas around the curves show the minimal and maximal values over the runs.

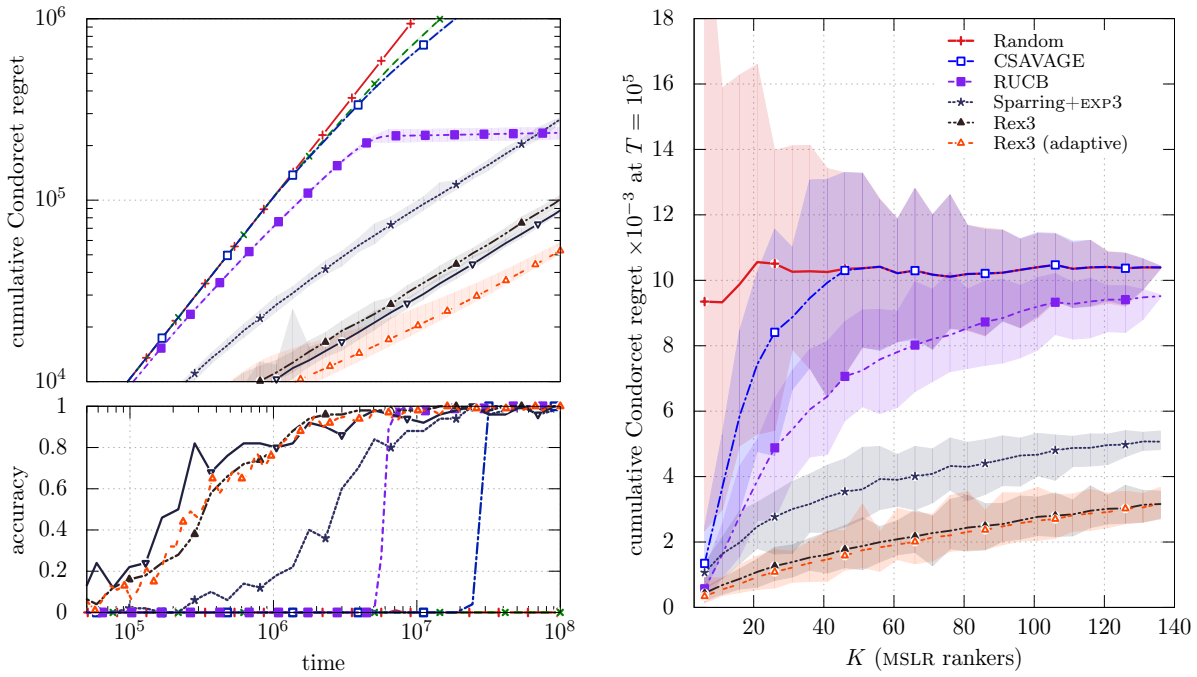


Figure 3. On the left: average regret and accuracy plots on MSLR30K with navigational queries ($K = 136$ rankers). On the right: same dataset, average regrets for a fixed $T = 10^5$ and K varying from 4 to 136.

The results of these experiments are summarized in Figure 2, and 3. Furthermore, similar experiments are given as extended material.

As expected, the adversarial-setting algorithms SPARRING and REX3 follow an $\mathcal{O}(\sqrt{T})$ regret curve while the stochastic-setting algorithms follow an $\mathcal{O}(\ln T)$ curve. Among the adversarial-setting algorithms, REX3 is shown to outperform SPARRING on all datasets. In the long run, adversarial-setting algorithms continue exploring and cannot compete in terms of regret against stochastic-setting algorithms, but the accuracy curves show that the cost of this exploration is very small. Moreover, for small horizons or high number of rankers, REX3 is extremely competitive against other algorithms.

This difference is clearly illustrated on the left-hand side of Figure 3 where we show the evolution of the expected cumulative regret at a fixed time horizon ($T = 10^5$) according to the number of arms. To obtain this plot we averaged the regret over 50 runs. For each K and each run we sampled uniformly K dimensions of the original 136×136 MSLR30K navigational preference matrix.

6. Conclusion

We proposed REX3, an exponential weighing algorithm for adversarial utility-based dueling bandits. We provided both an upper and a lower bound for its expected cumulative regret. These two bounds match the original bounds of the classical EXP3 algorithm. A thorough empirical study on several information retrieval datasets has confirmed the validity of these theoretical results. It also showed that REX3 and especially its anytime version with adaptive γ are competitive solutions for dueling bandits, even when compared to stochastic-setting algorithms in a stochastic environment.

A. Proof Sketch for Theorem 1

The general structure of the proof is similar to the one of (Auer et al., 2002b, section 3), but, as explained before, the $\hat{c}_i(t)$ estimator we use differs from the one of EXP3 because it gives an instantaneous regret estimate instead of an absolute reward estimate. As such, it may reach negative values and the $w_i(t)$ weights may decrease with time. We only give here a sketch of proof, stressing on the differences from (Auer et al., 2002b). An unfamiliar reader may refer to the extended version for step-by-step details.

Proof. Let $W_t = w_1(t) + w_2(t) + \dots + w_K(t)$. As in EXP3 proof we consider:

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^K \frac{p_i(t) - \gamma/K}{1 - \gamma} e^{(\gamma/K)\hat{c}_i(t)}$$

The inequality $e^x \leq 1 + x + (e-2)x^2$ is tight for $x \in [0, 1]$ but it remains valid for negative values, hence:

$$\begin{aligned} \frac{W_{t+1}}{W_t} &\leq 1 - \frac{\gamma^2/K}{1 - \gamma} \underbrace{\left(\frac{1}{K} \sum_{i=1}^K \hat{c}_i(t) \right)}_{=-M_1} \\ &\quad + \frac{(e-2)\gamma^2/K}{1 - \gamma} \underbrace{\left(\frac{1}{K} \sum_{i=1}^K p_i(t)\hat{c}_i(t)^2 \right)}_{=M_2} \end{aligned}$$

As in EXP3 we take the logarithm and sum over t . We get for any j :

$$\sum_{t=1}^T \frac{\gamma}{K} \hat{c}_j(t) - \ln(K) \leq \frac{\gamma^2/K}{1 - \gamma} M_1 + \frac{(e-2)\gamma^2/K}{1 - \gamma} M_2$$

By taking the expectation over the algorithm's randomization, we obtain for any j :

$$\begin{aligned} \sum_{t=1}^T \frac{\gamma}{K} \underbrace{\mathbb{E}_{\sim p} \hat{c}_j(t)}_{(8)} - \ln(K) &\leq \\ \frac{\gamma^2/K}{1 - \gamma} \sum_{i=t}^T \underbrace{\mathbb{E}_{\sim p} M_1}_{(9)} + \frac{(e-2)\gamma^2/K}{1 - \gamma} \sum_{i=t}^T \underbrace{\mathbb{E}_{\sim p} M_2}_{(10)} &\quad (7) \end{aligned}$$

From Lemma 1 we directly get the expected regret against j on the left side of the inequality:

$$\mathbb{E}_{\sim p} \hat{c}_j(t) = x_j - \mathbb{E}_{\sim p}(x_a) \quad (8)$$

By averaging (8) over the arms, we obtain:

$$\mathbb{E}_{\sim p(t)} M_1 = -\frac{1}{K} \sum_{i=1}^K \mathbb{E}_{\sim p} \hat{c}_i(t) = \mathbb{E}(x_a) - \frac{1}{K} \sum_{i=1}^K x_i \quad (9)$$

The following result is detailed in the extended version:

$$\mathbb{E}_{\sim p(t)} M_2 \leq \frac{1}{2} \mathbb{E}(x_a) + \frac{1}{2K} \sum_{i=1}^K x_i \quad (10)$$

From Lemma 1, (9), (10), and by definition of \mathbb{G}_{max} , $\mathbb{E}\mathbb{G}_{alg}$, and $\mathbb{E}\mathbb{G}_{unif}$, the inequality (7) rewrites into:

$$\begin{aligned} \mathbb{G}_{max} - \mathbb{E}\mathbb{G}_{alg} - \frac{K \ln K}{\gamma} &\leq \frac{\gamma}{1 - \gamma} (\mathbb{E}\mathbb{G}_{alg} - \mathbb{E}\mathbb{G}_{unif}) \\ &\quad + \frac{(e-2)\gamma}{2(1 - \gamma)} (\mathbb{E}\mathbb{G}_{alg} + \mathbb{E}\mathbb{G}_{unif}) \end{aligned}$$

Assuming $\gamma \leq \frac{1}{2}$ we finally obtain:

$$\mathbb{G}_{max} - \mathbb{E}\mathbb{G}_{alg} \leq \frac{K \ln K}{\gamma} + \gamma (e\mathbb{E}\mathbb{G}_{alg} - (4-e)\mathbb{E}\mathbb{G}_{unif})$$

□

Acknowledgments

We would like to thank the reviewers for their constructive comments and Masrour Zoghi who kindly sent us his experiment datasets.

References

- Ailon, N., Karnin, Z. S., and Joachims, T. (2014). Reducing dueling bandits to cardinal bandits. In *ICML 2014*, volume 32 of *JMLR Proceedings*, pages 856–864.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256.
- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.
- Bartók, G. (2013). A near-optimal algorithm for finite partial-monitoring games against adversarial opponents. In *Proc. COLT*.
- Bartók, G., Foster, D. P., Pál, D., Rakhlin, A., and Szepesvári, C. (2014). Partial monitoring - classification, regret bounds, and algorithms. *Math. Oper. Res.*, 39(4):967–997.
- Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.
- Busa-Fekete, R. and Hüllermeier, E. (2014). A survey of preference-based online learning with bandit algorithms. In *ALT 2014*, pages 18–39. Springer.
- Busa-Fekete, R., Hüllermeier, E., and Szörényi, B. (2014). Preference-based rank elicitation using statistical models: The case of mallows. In *ICML 2014*, *JMLR Proceedings*, pages 1071–1079.
- Busa-Fekete, R., Szörényi, B., Cheng, W., Weng, P., and Hüllermeier, E. (2013). Top-k selection based on adaptive sampling of noisy preferences. In *ICML 2013*, volume 28 of *JMLR Proceedings*, pages 1094–1102.
- Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA.
- Cesa-Bianchi, N. and Lugosi, G. (2009). Combinatorial bandits. In *COLT 2009*, pages 237–246. Omnipress.
- Chapelle, O., Joachims, T., Radlinski, F., and Yue, Y. (2012). Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.*, 30(1):6.
- Charon, I. and Hudry, O. (2010). An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals OR*, 175(1):107–158.
- Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969.
- Freund, Y. and Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103.
- Fürnkranz, J. and Hüllermeier, E., editors (2010). *Preference Learning*. Springer-Verlag.
- Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., and Gay, G. (2007). Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2).
- Karp, R. M. and Kleinberg, R. (2007). Noisy binary search and its applications. In *SODA 2007*, *SIAM Proceedings*, pages 881–890.
- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331.
- Liu, T.-Y., Xu, J., Qin, T., Xiong, W., and Li, H. (2007). LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR 2007*. ACM.
- MSLR30K (2012). Microsoft learning to rank dataset.
- Piccolboni, A. and Schindelhauer, C. (2001). Discrete prediction games with arbitrary feedback and loss. In *COLT/EuroCOLT*, volume 2111 of *LNCS*, pages 208–223. Springer.
- Radlinski, F. and Joachims, T. (2007). Active exploration for learning rankings from clickthrough data. In *KDD 2007*, pages 570–579. ACM.
- Seldin, Y., Szepesvári, C., Auer, P., and Abbasi-Yadkori, Y. (2012). Evaluation and analysis of the performance of the exp3 Algorithm in stochastic environments. In *EWRL*, volume 24 of *JMLR Proceedings*, pages 103–116.
- Urvoy, T., Clerot, F., Féraud, R., and Naamane, S. (2013). Generic exploration and K-armed voting bandits. In *ICML 2013*, volume 28 of *JMLR Proceedings*, pages 91–99.
- Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. (2012). The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556.
- Yue, Y. and Joachims, T. (2009). Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML 2009*, pages 1201–1208. Omnipress.

- Yue, Y. and Joachims, T. (2011). Beat the mean bandit. In *ICML 2011*, pages 241–248. Omnipress.
- Zoghi, M., Whiteson, S., and de Rijke, M. (2015). MergeRUCB: A method for large-scale online ranker evaluation. In *WSDM 2015*. ACM.
- Zoghi, M., Whiteson, S., Munos, R., and de Rijke, M. (2014a). Relative upper confidence bound for the k-armed dueling bandit problem. In *ICML 2014*, volume 32 of *JMLR Proceedings*, pages 10–18.
- Zoghi, M., Whiteson, S. A., de Rijke, M., and Munos, R. (2014b). Relative confidence sampling for efficient online ranker evaluation. In *WSDM 2014*, pages 73–82. ACM.