Relative Upper Confidence Bound for the K-Armed Dueling Bandit Problem

Zoghi, M.; Whiteson, S.; Munos, R.; de Rijke, M.

Published in:
JMLR Workshop and Conference Proceedings

Citation for published version (APA):
Relative Upper Confidence Bound for the \( K \)-Armed Dueling Bandit Problem

Masrour Zoghi\(^1\), Shimon Whiteson\(^1\)
Remi Munos\(^2\)
Maarten de Rijke\(^1\)
\(^1\)ISLA, University of Amsterdam, Netherlands
\(^2\)INRIA Lille - Nord Europe / MSR-NE

Abstract

This paper proposes a new method for the \( K \)-armed dueling bandit problem, a variation on the regular \( K \)-armed bandit problem that offers only relative feedback about pairs of arms. Our approach extends the Upper Confidence Bound algorithm to the relative setting by using estimates of the pairwise probabilities to select a promising arm and applying Upper Confidence Bound with the winner as a benchmark. We prove a sharp finite-time regret bound of order \( O(K \log T) \) on a very general class of dueling bandit problems that matches a lower bound proven in (Yue et al., 2012). In addition, our empirical results using real data from an information retrieval application show that it greatly outperforms the state of the art.

1. Introduction

In this paper, we propose and analyze a new algorithm, called Relative Upper Confidence Bound (RUCB), for the \( K \)-armed dueling bandit problem (Yue et al., 2012), a variation on the \( K \)-armed bandit problem in which the feedback comes in the form of pairwise preferences. We assess the performance of this algorithm using one of the main current applications of the \( K \)-armed dueling bandit problem, ranker evaluation (Joachims, 2002; Yue & Joachims, 2011; Hofmann et al., 2013a), which is used in information retrieval, ad placement and recommender systems, among others.

The \( K \)-armed dueling bandit problem is part of the general framework of preference learning (Fürnkranz & Hüllermeier, 2010), where the goal is to learn, not from real-valued feedback, but from relative feedback, which specifies only which of two alternatives is preferred. Developing effective preference learning methods is important for dealing with domains in which feedback is much more reliable if given in the form of a comparison (e.g., when provided by a human) and specifying real-valued feedback instead would be arbitrary or inefficient.

Other algorithms proposed for this problem are Interleaved Filter (IF) (Yue et al., 2012), Beat the Mean (BTM) (Yue & Joachims, 2011), and SAVAGE (Urvoy et al., 2013). All of these methods were designed for the finite-horizon setting, in which the algorithm requires as input the exploration horizon, \( T \), the time by which the algorithm needs to produce the best arm. The algorithm is then judged based upon either the accuracy of the returned best arm or the regret accumulated in the exploration phase.\(^1\) All three of these algorithms use the exploration horizon to set their internal parameters so that, for each \( T \), there is a separate algorithm IF\(_T\), BTM\(_T\) and SAVAGE\(_T\). By contrast, RUCB does not require this input, making it more useful in practice, since a good exploration horizon is often difficult to guess. Nonetheless, RUCB outperforms these algorithms in terms of the accuracy and regret metrics used in the finite-horizon setting.

The main idea of RUCB is to maintain optimistic estimates of the probabilities of all possible pairwise outcomes, and (1) use these estimates to select a potential champion, which is an arm that has a chance of being the best arm, and (2) select an arm to compare to this potential champion by performing regular Upper Confidence Bound (Agrawal, 1995) relative to it.

We prove a finite-time high-probability bound of \( O(K \log T) \) on the cumulative regret of RUCB, from which we deduce a bound on the expectation and all higher moments of cumulative regret. These bounds rely on substantially less restrictive assumptions on the \( K \)-armed dueling bandit problem than IF and BTM and have better multiplicative constants than those of SAVAGE. Further-

\(^1\)These terms are formalized in Section 2.
more, our bounds are the first explicitly non-asymptotic results for the $K$-armed dueling bandit problem.

More importantly, the main distinction of our result is that it holds for all time-steps. By contrast, given an exploration horizon $T$, the results for IF, BTM and SAVAGE bound only the regret accumulated by IF$_T$, BTM$_T$ and SAVAGE$_T$ in the first $T$ time-steps.

Finally, we evaluate our method empirically using real data from an information retrieval application. The results show that RUCB can learn quickly and effectively and greatly outperforms BTM and SAVAGE.

The main contributions of this paper are as follows:

- A novel algorithm for the $K$-armed dueling bandit problem that is more broadly applicable than existing algorithms,
- Regret bounds that make significantly less restrictive assumptions than IF and BTM, have better multiplicative constants than the results of SAVAGE, apply to all time-steps, and match an existing asymptotic lower bound,
- A novel proof technique that allows us to obtain the first logarithmic high probability regret bound for a UCB-type algorithm that does not require the probability of failure to be passed to the algorithm as a parameter: as a corollary, we also get the first logarithmic bounds on all higher moments of the cumulative regret for all times, and
- Experimental results, based on a real-world application, demonstrating the superior performance of our algorithm compared to existing methods.

2. Problem Setting

The $K$-armed dueling bandit problem (Yue et al., 2012) is a modification of the $K$-armed bandit problem (Thompson, 1933): the latter considers $K$ arms $\{a_1, \ldots, a_K\}$ and at each time-step, an arm $a_i$ can be pulled, generating a reward drawn from an unknown stationary distribution with expected value $\mu_i$. The $K$-armed dueling bandit problem is a variation in which, instead of pulling a single arm, we choose a pair $(a_i, a_j)$ and receive one of them as the better choice, with the probability of $a_i$ being picked equal to an unknown constant $p_{ij}$ and that of $a_j$ equal to $p_{ji} = 1 - p_{ij}$. We define the preference matrix $P = [p_{ij}]$, whose $ij$ entry is equal to $p_{ij}$.

In this paper, we assume that there exists a Condorcet winner (Urvoy et al., 2013): an arm, which without loss of generality we label $a_1$, such that $p_{ij} > \frac{1}{2}$ for all $i > 1$. Given a Condorcet winner, we define regret for each time-step as follows (Yue et al., 2012): if arms $a_i$ and $a_j$ were chosen for comparison at time $t$, then regret at that time is $r_t := \frac{\Delta_i + \Delta_j}{2}$, with $\Delta_k := p_{1k} - \frac{1}{2}$ for all $k \in \{1, \ldots, K\}$. Thus, regret measures the average advantage that the Condorcet winner has over the two arms being compared against each other. Given our assumption on the probabilities $p_{1k}$, this implies that $r = 0$ if and only if the best arm is compared against itself. We define cumulative regret up to time $T$ to be $R_T := \sum_{t=1}^T r_t$.

The goal of a bandit algorithm can be formalized in several ways. We consider two standard settings:

1. The finite-horizon setting, in which the algorithm is told in advance the exploration horizon, $T$, i.e., the number of time-steps that the evaluation process is given to explore before it has to produce a single arm as the best, which will be exploited thenceforth. In this setting, the algorithm can be assessed on its accuracy, the probability that a given run of the algorithm reports the Condorcet winner as the best arm (Urvoy et al., 2013), which is related to expected simple regret: the regret associated with the algorithm’s choice of the best arm, i.e., $r_{T+1}$ (Bubeck et al., 2009). Another measure of success in this setting is the amount of regret accumulated during the exploration phase, as used in the explore-then-exploit problem formulation (Yue et al., 2012).

2. The horizonless setting, in which no horizon is specified and the evaluation process continues indefinitely. Thus, it is no longer sufficient for the algorithm to maximize accuracy or minimize regret after a single horizon is reached. Instead, it must minimize regret across all horizons by rapidly decreasing the frequency of comparisons involving suboptimal arms, particularly those that fare worse in comparison to the best arm. This goal can be formulated as minimizing the cumulative regret over time, rather than with respect to a fixed horizon (Lai & Robbins, 1985).

All existing $K$-armed dueling bandit methods target the finite-horizon setting. However, we argue that the horizonless setting is more relevant in practice for the following reason: finite-horizon methods require a horizon as input and often behave differently for different horizons. This poses a practical problem because it is typically difficult to know in advance how many comparisons are required to determine the best arm with confidence and thus how to set the horizon. If the horizon is set too long, the algorithm is too exploratory, increasing the number of evaluations needed to find the best arm. If it is set too short, the best arm remains unknown when the horizon is reached and the algorithm must be restarted with a longer horizon.

Moreover, any algorithm that can deal with the horizonless setting can easily be modified to address the finite-horizon setting by simply stopping the algorithm when it reaches the horizon and returning the best arm. By contrast, for the reverse direction, one would have to resort to the “dou-
the number of arms is of moderate size. Moreover, one version of SAVAGE, called Condorcet SAVAGE, makes the Condorcet assumption and has the best theoretical results among the algorithms studied in that paper (Urvoy et al., 2013, Theorem 3). However, the regret bounds provided for Condorcet SAVAGE are of the form $O(K^2 \log T)$, and so are not as tight as those of IF, BTM or our algorithm.

Finally, note that all of the above results bound only $R_T$, where $T$ is the predetermined horizon, since IF, BTM and SAVAGE were designed for the finite-horizon setting. By contrast, in Section 5, we bound the cumulative regret of RUCB for all time-steps.

### 4. Method

**Algorithm 1 Relative Upper Confidence Bound**

- **Input:** $\alpha > \frac{1}{2}$, $T \in \{1, 2, \ldots\} \cup \{\infty\}$
- **Output:** $a_c$.

1. $W = [w_{ij}] = 0_{K \times K}$ // 2D array of wins: $w_{ij}$ is the number of times $a_i$ beat $a_j$.
2. $B = \emptyset$.
3. for $t = 1, \ldots, T$ do
   - $\mathbf{U} := [u_{ij}] = \frac{W}{W + \tau} + \sqrt{\frac{\alpha \ln t}{W + \tau}}$ // All operations are element-wise; $\tau := 1$ for any $x$.
   - $u_{ii} \leftarrow \frac{1}{2}$ for each $i = 1, \ldots, K$.
   - $C \leftarrow \{a_c | \forall j : u_{cj} \geq \frac{1}{2}\}$.
   - if $C = \emptyset$, then pick $c$ randomly from $\{1, \ldots, K\}$.
   - $B \leftarrow B \cap C$.
   - if $|C| = 1$, then $B \leftarrow C$ and $a_c$ be the unique element in $C$.
4. Sample $a_c$ from $C$ using the distribution:
   $$p(a_c) = \begin{cases} 0.5 & \text{if } a_c \in B, \\ \frac{1}{2^{\min(1, |B|)}} & \text{otherwise.} \end{cases}$$
5. **end if**
6. $d \leftarrow \arg \max_{i,j} u_{ij}$, with ties broken randomly.
7. Moreover, if there is a tie, $d$ is not allowed to be equal to $c$.
8. Compare arms $a_c$ and $a_d$ and increment $w_{cd}$ or $w_{dc}$ depending on which arm wins.
9. **end for**

**Return:** An arm $a_c$ that beats the most arms, i.e., $c$ with the largest count $\# \{j | \frac{w_{cj}}{w_{ij} + w_{cj}} > \frac{1}{2}\}$.

We now introduce Relative Upper Confidence Bound (RUCB), which is applicable to any $K$-armed dueling bandit problem with a Condorcet winner. In each time-step, RUCB, shown in Algorithm 1, goes through the following three stages:

1. RUCB puts all arms in a pool of potential champions. Then, it compares each arm $a_i$ against all other arms op-
timistically: for all $i \neq j$, it computes the upper bound $u_{ij}(t) = \mu_{ij}(t) + c_{ij}(t)$, where $\mu_{ij}(t)$ is the frequentist estimate of $p_{ij}$ at time $t$ and $c_{ij}(t)$ is an optimism bonus that increases with $t$ and decreases with the number of comparisons between $i$ and $j$ (Line 4). If $u_{ij} < \frac{1}{2}$ for any $j$, then $a_i$ is removed from the pool: the set of remaining arms is called $\mathcal{C}$. If we are left with a single potential champion at the end of this process, we let $a_c$ be that arm and put it in the set $\mathcal{B}$ of the hypothesized best arm (Line 9). Note that $\mathcal{B}$ is always either empty or contains one arm; moreover, an arm is demoted from its status as the hypothesized best arm as soon as it optimistically loses to another arm (Line 8). Next, from the remaining potential champions, a champion arm $a_c$ is chosen in one of two ways: if $\mathcal{B}$ is empty, we sample an arm from $\mathcal{C}$ uniformly randomly; if $\mathcal{B}$ is non-empty, the probability of picking the arm in $\mathcal{B}$ is set to $\frac{1}{2}$ and the remaining arms are given equal probability of being chosen (Line 11).

(2) Regular UCB is performed using $a_c$ as a benchmark (Line 13), i.e., UCB is performed on the set of arms $a_1, \ldots, a_K$. Specifically, we select the arm $d = \arg \max_j u_{jc}$. When $c \neq j$, $u_{jc}$ is defined as above. When $c = j$, since $p_{cc} = \frac{1}{2}$, we set $u_{cc} = \frac{1}{2}$ (Line 5).

(3) The pair $(a_c, a_d)$ is compared and the score sheet is updated as appropriate (Line 7).

Note that in stage (1) the comparisons are based on $u_{ij}$, i.e., $a_c$ is compared optimistically to the other arms, making it easier for it to become the champion. By contrast, in stage (2) the comparisons are based on $u_{jc}$, i.e., $a_c$ is compared to the other arms pessimistically, making it more difficult for $a_c$ to be compared against itself. This is important because comparing an arm against itself yields no information. Thus, RUCB strives to avoid auto-comparisons until there is great certainty that $a_c$ is indeed the Condorcet winner.

Eventually, as more comparisons are conducted, the estimates $\mu_{ij}$ tend to concentrate above $\frac{1}{2}$ and the optimism bonuses $c_{ij}$ become small. Thus, both stages of the algorithm increasingly select $a_1$, i.e., $a_c = a_d = a_1$, which accumulates zero regret.

Note that Algorithm 1 is a finite-horizon algorithm if $T < \infty$ and a horizonless one if $T = \infty$, in which case the for loop never terminates.

5. Theoretical Results

In this section, we prove finite-time high-probability and expected regret bounds for RUCB. We first state Lemma 1 and use it to prove a high-probability bound on the number of comparisons for each suboptimal arm in Proposition 2. An immediate consequence of this result is a high probability regret bound of the form $O(K^2 \log T)$, which is similar to the bound for SAVAGE (Urvoy et al., 2013) but for the horizonless setting. However, in Theorem 4 we show that this can be lowered to $O(K \log T)$ and we deduce an expected regret bound in Theorem 5. This result is proven under conditions that are much more general than those for IF (Yue et al., 2012) and without requiring the user to specify the $\gamma$ parameter as BTM does (Yue & Joachims, 2011). Moreover, it matches the asymptotic lower bound proven in (Yue et al., 2012, Theorem 2).

The results in Theorems 4 and 5 are surprising because a $K$-armed dueling bandit problem depends on roughly $K^2$ independent parameters, so one would expect a bound of the form $O(K^2 \log T)$ unless strong prior information is infused into the algorithm, as with IF and BTM. However, these theorems show that one can get asymptotic behaviour resembling that of a regular $K$-armed bandit algorithm on a very broad class of dueling bandit problems with very little prior knowledge. This finding is also of great practical significance because there are many situations in which one has a choice between applying a $K$-armed bandit algorithm to an unreliable quantity, such as Click Through Rate, or using a $K$-armed dueling bandit algorithm to conduct direct comparisons, which are known to be more reliable when dealing with humans (Hofmann et al., 2013b, §2.1). These results show that, given such a dilemma, using a dueling bandit approach does not come at the expense of the asymptotic behaviour.

Finally, note that the high probability bound proven in Theorem 4 does not rely on the probability of failure, $\delta$, being passed to the algorithm. Thus, we can use it to also bound higher moments (hence also the variance) of the cumulative regret for RUCB for all times. This is in contrast to high probability bounds that require $\delta$ to be specified before the algorithm starts (Audibert et al., 2009; Srinivas et al., 2010; Abbasi-yadkori et al., 2011), from which one cannot obtain expected regret bounds for all times. While, given a time $T$, one can set $\delta = 1/T$ in the algorithm to get a logarithmic expected regret bound at time $T$, getting a logarithmic expected regret bound at time $T^{1+\epsilon}$ for any $\epsilon > 0$, requires rerunning the algorithm with $\delta = 1/T^{1+\epsilon}$.

As before, we assume without loss of generality that $a_1$ is the optimal arm. See Table 1 for definitions of symbols used throughout.

Lemma 1. Let $\mathbf{P} := [p_{ij}]$ be the preference matrix of a $K$-armed dueling bandit problem with arms $\{a_1, \ldots, a_K\}$. Then, for any dueling bandit algorithm and any $\alpha > \frac{1}{2}$ and $\delta > 0$, we have

$$P\left(\forall t > C(\delta), i, j, p_{ij} \in [u_{ij}(t), u_{ij}(t)]\right) > 1 - \delta.$$ 

Proof. See §8.2 in the supplementary material.
against each other, we must have compared against each other more than now, let us assume that distinct arms Condorcet winner we have a

\[
P \left( \exists t, (i,j) \neq (1,1): N_{ij}(t) > C(\delta) \vee D_{ij} \ln t \right) < \delta \quad (1)
\]

and, \(N_{ij}^d(t)\), the number of times \(a_i\) was compared against \(a_j\) between time-steps \(C(\delta)\) and \(t\), satisfies

\[
P \left( \exists t > C(\delta), (i,j) \neq (1,1): N_{ij}^d(t) > D_{ij} \ln t \right) < \delta \quad (2)
\]

**Proof.** Given Lemma 1, we know with probability \(1 - \delta\) that \(p_{ij} \in \left[ u_{ij}(t), \frac{N_{ij}(t)}{t} \right]\) for all \(t > C(\delta)\). Let us first deal with the easy case when \(i = j \neq 1\): when \(t > C(\delta)\) holds, \(a_i\) cannot be played against itself, since if we get \(c = i\) in Algorithm 1, then by Lemma 1 and the fact that \(a_1\) is the Condorcet winner we have \(d \neq i\) since \(u_{ii}(t) = \frac{1}{2} < p_{ii} \leq u_{ij}(t)\).

Now, let us assume that distinct arms \(a_i\) and \(a_j\) have been compared against each other more than \(D_{ij} \ln t\) times and that \(t > C(\delta)\). If \(s\) is the last time \(a_i\) and \(a_j\) were compared against each other, we must have

\[
u_{ij}(s) - l_{ij}(s) = 2 \sqrt{\frac{\alpha \ln s}{N_{ij}(t)}} \leq 2 \sqrt{\frac{\alpha \ln t}{\min\{\Delta_i, \Delta_j\}}} = \min\{\Delta_i, \Delta_j\}. \quad (3)
\]

On the other hand, for \(a_i\) to have been compared against \(a_j\) at time \(s\), one of the following two scenarios must have happened:

1. In Algorithm 1, we had \(c = i\) and \(d = j\), in which case both of the following inequalities must hold:

   a. \(u_{ij}(s) \geq \frac{1}{2}\), since otherwise \(c\) could not have been set to \(i\) by Line 5 of Algorithm 1, and

   b. \(l_{ij}(s) = 1 - u_{ij}(s) \leq 1 - p_{ii} = p_{ij}\), since we know that \(p_{ii} \leq u_{ij}(t)\), by Lemma 1 and the fact that \(t > C(\delta)\), and for \(d = j\) to be satisfied, we must have \(u_{ij}(t) \leq u_{ji}(t)\) by Line 6 of Algorithm 1.

   From these two inequalities, we can conclude

   \[
u_{ij}(s) - l_{ij}(s) \geq \frac{1}{2} - p_{ii} = \Delta_i. \quad (4)
\]
This inequality is illustrated using the lower right confidence interval in the \((a_1, a_2)\) block of Figure 1, where the interval shows \([l_{ij}(s), u_{ij}(s)]\) and the distance between the dotted lines is \(\frac{1}{2} - p_{j1}\).

II. In Algorithm 1, we had \(c = j\) and \(d = i\), in which case swapping \(i\) and \(j\) in the above argument gives

\[
u_{ji}(s) - l_{ji}(s) \geq \frac{1}{2} - p_{j1} = \Delta_j.
\] (5)

Similarly, this is illustrated using the lower left confidence interval in the \((a_1, a_i)\) block of Figure 1, where the interval shows \([l_{ji}(s), u_{ji}(s)]\) and the distance between the dotted lines is \(\frac{1}{2} - p_{i1}\).

Putting (4) and (5) together with (3) yields a contradiction, so with probability \(1 - \delta\) we cannot have \(N_{ij}\) be larger than both \(C(\delta)\) and \(D_{ij} \ln t\). This gives us both (1) and (2). \(\square\)

We use the next definition in what follows:

**Definition 3.** Let \(T_\delta\) be the smallest time satisfying

\[
\hat{T}_\delta > C \left(\frac{\delta}{2}\right) + \sum_{j < i} D_{ij} \ln \hat{T}_\delta,
\]

which is guaranteed to exist since the expression on the left of the inequality grows linearly with \(T_\delta\) and the expression on the right grows logarithmically. Note that \(T_\delta\) is specified by the \(K\)-armed bandit problem.

With this in hand, we now state our main result:

**Theorem 4.** Given the setup of Proposition 2, for any \(\delta > 0\), we have with probability \(1 - \delta\) that for all times \(T\) the following bound on the cumulative regret holds:

\[
R_T \leq \hat{C}(\delta) + \sum_{j=2}^{K} \hat{D}_j \ln T,
\] (6)

where

\[
\hat{C}(\delta) := \left(4 \ln \frac{2}{\delta} + 2C \left(\frac{\delta}{2}\right) + 2D \ln 2D\right) \Delta_{\text{max}}^2,
\]

\[
\hat{D}_j := D_{ij} \left(\Delta_{ij} + 2\Delta_{\text{max}}\right) = \frac{2C (\Delta_j + 4\Delta_{\text{max}})}{\Delta_j^2},
\]

with \(C(\cdot)\) and \(D\) as in Proposition 2, and \(\Delta_{\text{max}} := \max_i \Delta_i\) and \(\Delta_{ij} := \frac{\Delta_i + \Delta_j}{2}\), while \(R_T\) is the cumulative regret as defined in Section 2.

**Proof.** If we apply Inequality (2) in Proposition 2 with \(t = \hat{T}_\delta\) (as in Definition 3), we know that with probability \(1 - \frac{\delta}{2}\) there is a time \(T_\delta \in \left(C \left(\frac{\delta}{2}\right), \hat{T}_\delta\right]\) when arm \(a_1\) was compared against itself, which means that at that time we had \(u_{i1}(T_\delta) < \frac{1}{2}\). This in turn implies that \(B = \{a_1\}\) from that point on, since by Lemma 1 we have that \(\frac{1}{2} < p_{i1} \leq u_{i1}(t)\) for all \(t > T_\delta > C \left(\frac{\delta}{2}\right)\).

Since we have \(B = \{a_1\}\), we know that when choosing \(a_c\) in Algorithm 1, the probability of choosing \(a_1\) is equal to \(\frac{1}{2}\). Given this, we can expect that from \(T_\delta\) onwards, the algorithm will spend roughly half of its time comparing \(a_1\) against other arms. In what follows, we show that this is indeed the case.

Let \(\hat{N}_{ij}(T)\) denote the number of times arm \(a_i\) was compared against arm \(a_j\) between times \(T_\delta\) and \(T\). Proposition 2 shows that, again with probability \(1 - \frac{\delta}{2}\), we have \(\hat{N}_{ij}(T) \leq D_{ij} \ln T\) for all \(i < j\); note that this \(1 - \frac{\delta}{2}\) is the same as the one used above. In particular, this means that \(\hat{N}_1(T)\), the number of times between times \(T_\delta\) and \(T\) when we had \(c = 1 \neq d\), is bounded by

\[
\hat{N}_1(T) \leq \sum_{j=2}^{K} \hat{N}_{ij}(T) \leq \sum_{j=2}^{K} D_{ij} \ln T =: \hat{N}_1(T).
\] (7)

Let us introduce here two sets of random variables:

- \(\tau_0, \tau_1, \tau_2, \ldots\), where \(\tau_0 = T_\delta\) and \(\tau_1\) is the \(t^{th}\) time arm \(a_1\) was compared against another arm after \(T_\delta\).
- \(n_1, n_2, \ldots\), where \(n_l\) is the number of times in Algorithm 1 we had \(c = 1 \neq d\) between \(\tau_{l-1}\) and \(\tau_l\).

Now, note that RUCB chooses \(c \neq 1\) or \(d \neq 1\) in time-step \(t\) if and only if \(u_{1j}(t) \geq \frac{1}{2}\) for some \(j \neq 1\) and that we can have \(u_{1j}(t+1) < u_{1j}(t)\) only if at the end of the \(t^{th}\) iteration, arm \(a_1\) was compared against arm \(a_j\). In other words, whenever we have \(u_{1j}(T) \geq \frac{1}{2}\) for some \(j > 1\), the algorithm will continue to set \((c, d) \neq (1, 1)\) until all of the \(u_{1j}\) with \(j > 1\) get submerged below \(\frac{1}{2}\) and that the last comparison before we get to this state must be between \(a_1\) and another arm. With this picture in mind, with probability \(1 - \frac{\delta}{2}\), we have

\[
R_T \leq T_\delta \Delta_{\text{max}} + \sum_{j=2}^{K} D_{ij} \Delta_{ij} \ln T + \sum_{i=1}^{K} n_i \Delta_{\text{max}},
\] (8)

where \(\hat{N}_1(T)\) is as in Inequality (7), and so all we need to do is bound \(T_\delta\) and the sum of the intervals \(n_l\) for \(l = 1, \ldots, \hat{N}_1(T)\). Let us deal with the former first: we know that \(T_\delta \leq T_{\hat{\delta}}\) and that the latter is defined to be the smallest time-step satisfying the inequality in Definition 3, so all we need to do is produce one number that, when plugged in for \(T_\delta\), satisfies the inequality, and one such number is \(2C \left(\frac{\delta}{2}\right) + 2D \ln 2D\). To see this, let us temporarily use the notation \(C := C \left(\frac{\delta}{2}\right)\), and use the concavity of the log function, a first order Taylor expansion, and the fact that we have \(\ln x < x\) for any \(x\), to get

\[
C + D \ln(2C + 2D \ln 2D)
\leq C + D \ln(2D \ln 2D) + \frac{2C}{2D} \ln 2D
\leq C + D \ln(2D)^2 + C = 2C + 2D \ln 2D,
\]

where we used the fact that \(D > 2\) and so \(\ln 2D > 1\).
Let us now return to the task of bounding the sum of the intervals \( n_j \). To do so, we introduce the random variables \( \tilde{n}_1, \tilde{n}_2, \ldots \), which are independent samples from the geometric distribution with decay \( \frac{1}{2} \). Note that \( \tilde{n}_1 \) bounds \( n_j \) from above since it counts the number of iterations it would take for Line 11 of Algorithm 1 to produce \( a_j \) and once we have \( c = 1 \), we are guaranteed to have a comparison between \( a_j \) and another arm, as long as \( u_j \geq \frac{1}{2} \) for some \( j > 1 \). Furthermore, the sum of independent geometric random variables has a negative binomial distribution (Feller, 1968, §VI.8), with the following probability mass function, cf. (Feller, 1968, Equation VI.8.1):

\[
f(n; r) := P \left( \sum_{l=1}^{r} \tilde{n}_l = n \right) = \frac{(n+r-1)}{2^{n+r}},
\]

where in our case \( p = \frac{1}{2} \) and so it is eliminated from the notation of the PMF. In order to bound this sum with high probability, we note that when \( n \geq 2r \), then we have

\[
f(n; r) = \frac{(n+r-1)!}{2^{n+r}} \leq (n+r-1) \quad n! (r-1)! (n+r)!
\]

\[
= 2(n+1) \quad 2 \left[ 1 - \frac{r-1}{n+r} \right] \geq 2 \frac{2r - 2}{3r} > 4.
\]

Thus, we have \( f(n; r) \leq (2r; r) \left( \frac{3}{4} \right)^{n-2r} \leq \left( \frac{3}{4} \right)^{n-2r} \) for all \( n \geq 2r \), since \( f(2r; r) \) is a probability and so at most equal to 1. From this we can conclude that with probability

\[
1 - \frac{\epsilon}{2}, \quad \text{we have} \quad n \leq 2r + \frac{\ln \frac{1}{\epsilon}}{\ln \frac{3}{4}}, \quad 2r - 4 \ln \frac{3}{4}.
\]

Next, we state our expected regret bound, which is a direct consequence of Theorem 4.

**Theorem 5.** Given the setup of Proposition 2 together with the notation of Theorem 4, we have the following expected regret bound for RUCB, where the expectations are taken across different runs of the algorithm: if we have \( \alpha > 1 \), the expected regret accumulated by RUCB after \( T \) iterations is bounded by

\[
E[R_T] \leq \left[ 8 + \left( 2(4\alpha - 1)K^2 \right) \left( \frac{2\alpha - 1}{2\alpha - 1} \right) \right] \Delta_{\max} + 2D \Delta_{\max} \ln 2D + \sum_{j=2}^{K} \frac{2\alpha (\Delta_j + 4\Delta_{\max})}{\Delta_j} \ln T,
\]

**Proof.** See §8.3 in the supplementary material.

**Remark 6.** (1) Using a very similar argument as the one used to prove Theorem 5, we can also bound the \( m^{th} \) moment of \( R_T \) whenever we have \( \alpha > \frac{m+1}{2} \), which can be used to bound its variance for \( \alpha > 1.5 \).

(2) In general, our regret bounds are not directly comparable to those of IF and BTM, since those bounds depend only on \( \Delta_{\max} \); so, if the majority of the \( \Delta_j \) are larger than \( \Delta_{\min} \), then our upper bound is lower than that of IF and BTM. On the other hand, if most \( \Delta_j \) are close to \( \Delta_{\min} \), then the upper bound for IF would be lower: the same would hold for BTM if \( \gamma \) is small.

(3) Note that RUCB uses the upper-confidence bounds (Line 3 of Algorithm 1) introduced in the original version of UCB (Auer et al., 2002) (up to the \( \alpha \) factor). Recently refined upper-confidence bounds (such as UCB-V (Audibert et al., 2009) or KL-UCB (Cappé et al., 2013)) have improved performance for the regular \( K \)-armed bandit problem. However, in our setting the arm distributions are Bernoulli and the comparison value is 1/2. Thus, since we have \( 2\Delta_j^2 \leq kl(p_{j1}/1/2) \leq 4\Delta_j^2 \) (where \( kl(a, b) = a \ln(1 + (1 - a) \ln \frac{1}{2} - b) \) is the KL divergence between Bernoulli distributions with parameters \( a \) and \( b \)), we deduce that using KL-UCB instead of UCB does not improve the leading constant in the logarithmic term of the regret by a numerical factor of more than 2.

### 6. Experiments

To evaluate RUCB, we apply it to the problem of ranker evaluation from the field of information retrieval (IR) (Manning et al., 2008). A ranker is a function that takes as input a user’s search query and ranks the documents in a collection according to their relevance to that query. Ranker evaluation aims to determine which among a set of rankers performs best. One effective way to achieve this is to use interleaved comparisons (Radlinski et al., 2008), which interleave the documents proposed by two different rankers and presents the resulting list to the user, whose resulting click feedback is used to infer a noisy preference for one of the rankers. Given a set of \( K \) rankers, the problem of finding the best ranker can then be modeled as a \( K \)-armed dueling bandit problem, with each arm corresponding to a ranker.

We evaluated RUCB, Condorcet SAVAGE and BTM using randomly chosen subsets from the pool of 64 rankers provided by LETOR, a standard IR dataset (see §8.4 for more details of the experimental setup), yielding \( K \)-armed dueling bandit problems with \( K \in \{16, 32, 64\} \). For each set of rankers, we performed 100 independent runs of each algorithm for a maximum of 4.5 million iterations. For RUCB we set \( \alpha = 0.51 \), which approaches the limit set by our high-probability result. Since BTM and SAVAGE
require the exploration horizon as input, we ran BTMₖ and CSAFEₖ for various horizons 𝑇 ranging from 1000 to 4.5 million. In the plots in Figure 2, the markers on the green and the blue curves show the regret accumulated by BTMₖ and CSAFEₖ in the first 𝑇 iteration of the algorithm for each of these horizons. Thus, each marker corresponds, not to the continuation of the runs that produced the previous marker, but to new runs conducted with a larger 𝑇.

Since RUCB is horizonless, we ran it for 4.5 million iterations and plotted the cumulative regret, as shown using the red curves in the plots in Figure 2. For all three algorithms, the middle curve shows average cumulative regret and the dotted lines show minimum and maximum cumulative regret across runs. Note that these plots are in log-linear scale, so they depict the relation between 𝑅𝑇 and 𝑇, which can be seen to be asymptotically linear. The regret curves for BTM are cut-off in these plots, since in all three experiments 𝑅〗_{𝐵𝑇𝑀} grew linearly with 𝑇 in the first 4.5 million iterations. As can be seen from the plots in Figure 2, RUCB accumulates the least regret of the three algorithms: the average regret accumulated by RUCB is less than half of that Condorcet SAVAGE by the end of each of the three experiments and even the worst performing run of RUCB accumulated considerably less regret than the best performing run of Condorcet SAVAGE.

7. Conclusions

This paper proposed a new method called Relative Upper Confidence Bound (RUCB) for the 𝐾-armed duel ing bandit problem that extends the Upper Confidence Bound (UCB) algorithm to the relative setting by using optimistic estimates of the pairwise probabilities to choose a potential champion and conducting regular UCB with the champion as the benchmark.

We proved finite-time high-probability and expected regret bounds for RUCB that match an existing lower bound. Unlike existing results, our regret bounds hold for all time-steps, rather than just a specific horizon 𝑇 input to the algorithm. Furthermore, they take the form 𝑂(𝐾 log 𝑇) while making much less restrictive assumptions than existing algorithms with similar bounds. Finally, the empirical results showed that RUCB greatly outperforms state-of-the-art methods.

In future work, we will consider two extensions to this research. First, building off extensions of UCB to the continuous bandit setting (Srinivas et al., 2010; Bubeck et al., 2011; Munos, 2011; de Freitas et al., 2012; Valko et al., 2013), we aim to extend RUCB to the continuous duel ing bandit setting, without a convexity assumption as in (Yue & Joachims, 2009; Jamieson et al., 2012). Second, building off Thompson Sampling (Thompson, 1933; Agrawal & Goyal, 2012; Kauffmann et al., 2012), an elegant and effective sampling-based alternative to UCB, we will investigate whether a sampling-based extension to RUCB would be amenable to theoretical analysis. Both these extensions involve overcoming not only the technical difficulties present in the regular bandit setting, but also those that arise from the two-stage nature of RUCB. Since the submission of this paper, the latter of these two ideas has been validated experimentally in (Zoghi et al., 2014), although a theoretical analysis is still lacking.

Acknowledgments

This research was partially supported by the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 270327, nr 288024 and nr 312827, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, the Center for Creation, Content and Technology (CCCT), the QuaMerdes project funded by the CLARIN-nl program, the TROVe project funded by the CLARIAH program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105 the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.
References


