Multi-Dueling Bandits and Their Application to Online Ranker Evaluation

Brian Brost[1], Yevgeny Seldin[1], Ingemar J. Cox[1,2], Christina Lioma[1]

[1] University of Copenhagen
[2] University College London

brian.brost@di.ku.dk, seldin@di.ku.dk, ingemar@ieee.org, c.lioma@di.ku.dk

ABSTRACT

Online ranker evaluation focuses on the challenge of effi-
ciently determining, from implicit user feedback, which ranker
out of a finite set of rankers is the best. It can be modeled
by *dueling bandits*, a mathematical model for online learning
under limited feedback from pairwise comparisons. Compar-
isons of pairs of rankers is performed by *interleaving* their
result sets and examining which documents users click on.
The dueling bandits model addresses the key issue of which
pair of rankers to compare at each iteration.

Methods for simultaneously comparing more than two rankers
have recently been developed. However, the question of
which rankers to compare at each iteration was left open.
We address this question by proposing a generalization of
the dueling bandits model that uses simultaneous compar-
isons of an unrestricted number of rankers.

We evaluate our algorithm on several standard large-scale
online ranker evaluation datasets. Our experimental results
show that the algorithm yields orders of magnitude improve-
ment in performance compared to state-of-the-art dueling
bandit algorithms.

1. INTRODUCTION

Evaluation of rankers can be done online by presenting
the ranked lists produced by rankers to users and then in-
ferring the quality of the rankers by analyzing users' clicks.
Online evaluation of rankers has become increasingly popu-
lar, partly because user behaviour can be easily logged. This
provides online evaluation methods with inexpensive access
to large amounts of training data [5]. One of the key draw-
backs of online evaluation methods is that the outputs of
new, potentially poor, rankers need to be presented to ac-
tual users. If a new ranker turns out to be poor, then users
will be presented with poor results and, in the worst case,
might abandon the service [7]. Conversely, if new rankers are
not presented there is a risk of overlooking better rankers in
the pool of rankers.

In online evaluation, it is usually easier for users to make
relative judgements, rather than absolute judgements. Rankers
can be compared by *interleaving* their result lists and exam-

ining which documents a user clicks on. Interleaving meth-
ods were found to require 1-2 orders of magnitude less data
than absolute metrics to detect even small differences in re-
trieval quality [5].

When using interleaving to compare pairs of rankers, it
is critical to determine which two rankers to interleave at
each comparison, i.e. to resolve the exploration-exploitation
tradeoff. *Dueling bandits* is an elegant mathematical frame-
work for solving this [15, 18, 19, 17].

Recently, interleaving has been generalized to *multileav-
ing* which permits more than two rankers to be compared
in a single comparison [12, 13, 2]. Prior work on multileav-
ing has shown that repetitive simultaneous comparison of
fixed sets of rankers through multileaving is similarly accu-
rate in identifying the best ranker in the set and faster than
sequential comparisons of pairs of rankers through interleav-
ing. However, prior work focused only on the comparisons
themselves, and did not address the key issue of selecting
subsets of rankers for each comparison. This approach has
disadvantages. Firstly, since poor rankers are participating
on all the comparisons the quality of the multileaved lists
throughout the evaluation process is poor. Secondly, very
poor rankers that could potentially be eliminated early in
the process continue being evaluated, which does not allow
the comparisons to focus on rankers whose quality is harder
to distinguish.

We extend the dueling bandit framework and propose
a *Multi-Dueling Bandit* algorithm that provides an intel-
ligent selection of rankers for simultaneous comparisons and
improves the trade-off between exploration and exploita-
tion. Our experimental evaluations of the new algorithm
on real web search learning-to-rank datasets show that our
algorithm yields orders of magnitude improvement in per-
formance compared to state-of-the-art dueling bandit algo-
rithms.

2. BACKGROUND AND RELATED WORK

Our work builds on two main bodies of literature. The
first is concerned with *selecting* promising rankers from a
pool of rankers and the second is concerned with *combining*
their ranked lists and *evaluating* their relative quality. The
corresponding prior work is discussed next.

2.1 Dueling Bandits

Learning from limited relative feedback from pairwise com-
parisons can be modeled as a K-armed *dueling* bandit prob-
lem [15]. The goal is to select pairs of rankers, so that a
mix of their rankings will be almost as good as the ranking

of the best ranker in the pool. There are several possible definitions of the best ranker for pairwise comparisons. The most common definition is the *Condorcet winner*, which is a ranker that is (pairwise) better that any other ranker in the pool. Note that a Condorcet winner is not guaranteed to exist. Nevertheless, many dueling bandit algorithms assume the existence of a Condorcet winner and we follow suit.

Zoghi *et al.* [18] proposed an algorithm for the dueling bandit setting based on the idea of *relative upper confidence bounds* (RUCB). The algorithm maintains a relative upper confidence bound on the probability that a given arm i is better than another arm j. The algorithm then selects an arm i that might be the best, based on its upper confidence bounds relative to all other arms, and then selects the challenger with the highest upper confidence bound relative to i. Zoghi *et al.* [17] subsequently proposed a divide-and-conquer algorithm, *MergeRUCB*, extending their earlier work in [18]. Experiments by Zoghi *et al.* suggest that for large numbers of arms it outperforms RUCB.

Komiyama *et al.* [10] proposed an algorithm, *Relative Minimum Empirical Divergence* (RMED), which draws arms based on whether an arm has not been compared with other arms sufficiently often, or if it is not substantially beaten by many other arms. To decide if an arm has been sufficiently explored it uses bounds based on the KL-divergence. They showed that this algorithm outperformed RUCB and MergeRUCB. To the best of our knowledge this is currently the best performing dueling bandit algorithm.

All algorithms listed above are limited to pairwise comparisons, whereas our proposed algorithm is based on simultaneous comparisons of multiple rankers.

2.2 Comparison Methods

Combining ranked lists and evaluating chosen rankers is based on click-through information obtained by *interleaving* (for two rankers) or *multileaving* (for two or more rankers) methods.

Interleaving has been generalised to simultaneous comparison of more than two rankers. The proposed multileaving methods include Team Draft Multileave (TDM) [12], Optimised Multileave (OM) [12], Probabilistic Multileave (PM) [13], and Sample Only Scored Multileave (SOSM) [2]. SOSM was found to outperform the other multileaving methods, which can either be less accurate, or fail to scale well with the number of rankers [2]. We used SOSM to multileave rankers for our multi-dueling bandit algorithm.

3. THE MULTI-DUELING BANDIT PROBLEM

In multi-dueling bandits, at each iteration, t, an algorithm selects a subset, S_t, of K arms and observes outcomes of noisy pairwise comparisons (duels) between all pairs of arms in S_t. In the ranking scenario this corresponds to multileaving the ranked lists of the subset, S_t, of rankers and then inferring the relative quality of the lists (and the corresponding rankers) from user clicks. When the size of S_t is limited to 2 the problem reduces to standard dueling bandits.

Let $P = [p_{ij}]$ be a matrix of probabilities that arm i wins in a pairwise comparison with arm j (it satisfies $p_{ij} = 1 - p_{ji}$ and we define $p_{ii} = \frac{1}{2}$). In pairwise comparisons the best arm is not always well-defined. We follow the assumption in most dueling bandit literature that there exists a Condorcet

winner, which is a unique arm $*$ satisfying $p_{*j} > \frac{1}{2}$ for all $j \neq *$. That is, the Condorcet winner $*$ is pairwise better than any other arm j. The quality of all arms is then defined by their *regret*, $r(j) = p_{*j} - \frac{1}{2}$, which is a shifted probability of losing to the best arm (this definition also coincides with dueling bandits). Smaller regret corresponds to better quality and the regret of playing the best arm is zero. The quality of a set of arms S_t is defined by the average quality of the constituent arms (the average regret)

$$r(S_t) = \frac{\sum_{j \in S_t} p_{*j}}{|S_t|} - \frac{1}{2}. \tag{1}$$

The goal of a multi-dueling bandit algorithm is to select subsets of arms $S_1, S_2, \ldots$, so that the cumulative regret $\sum_{t=1}^{T} r(S_t)$ is minimized. All arms have to be selected a small number of times in order to be explored, but the goal of the algorithm is to minimize the number of times when suboptimal arms are selected.

Simultaneous comparison of more than two arms may affect their pairwise winning probabilities. For example, in ranking, the effective length of a multileaved ranked list is typically limited by 10 items, since users rarely go beyond the first page of results. Therefore, the simultaneous comparison of more than 10 rankers means that some rankers may be compared based on a merged list that does not include their top suggestions. This may affect the estimates of their relative quality. This effect, which we refer to as *distortion* may also occur when less than 10 rankers are compared, since the limited length of the merged list does not allow perfect representation of every ranker. The exact level of distortion depends on the data, ranker, and method used for multileaving. The level of distortion of estimates of the pairwise winning probabilities made by SOSM, which was used in our experiments, is evaluated in Section 4.2.4. It is important to emphasize that in all our experimental comparisons, except one pathological case, the advantage of parallel exploration outweighed the disadvantage due to distortion in estimates.

3.1 Multi-Dueling Bandit Algorithm

The proposed multi-dueling bandit algorithm maintains optimistic estimates of pairwise winning probabilities p_{ij} and plays arms that, according to these optimistic estimates, have a chance of being the Condorcet winner. When there is a single candidate, the algorithm *exploits* this knowledge and plays only that candidate. When there are multiple candidates the algorithm *explores* by comparing them all. We increase parallel exploration by adding additional arms to such comparisons, as described below.

Our estimates of pairwise winning probabilities are based on empirical counts of wins/losses. In order for these estimates to be meaningful the algorithm has to assume that pairwise winning probabilities are *consistent* with the pairwise winning probability matrix P, irrespective of the composition of the set S_t (meaning that they are not distorted). More precisely, since correct identification of the Condorcet winner depends on correct estimation of the probabilities p_{*j}, it is important that they remain at a certain margin above $\frac{1}{2}$ irrespective of the composition of S_t. Incorrect estimation of p_{ij}-s for $i, j \neq *$ does not influence identification of the Condorcet winner and, therefore, their distortion does not disturb the operation of the algorithm.

We now describe our algorithm, which is provided in the

Algorithm 1 box. We denote by $n_{ij}(t)$ the number of times up to round t that i and j were compared with each other. Let $w_{ij}(t)$ denote the number of times when arm i beat arm j. We break ties randomly, so that $n_{ij}(t) = w_{ij}(t) + w_{ji}(t)$. We compute upper confidence bounds $u_{ij}(t)$ on the probabilities p_{ij}:

$$u_{ij}(t) = \frac{w_{ij}(t)}{n_{ij}(t)} + \sqrt{\frac{\alpha \ln t}{n_{ij}(t)}} \qquad (2)$$

(u_{ij}-s are the optimistic estimates of p_{ij}-s and they are analogous to those used in [18] for pairwise comparisons). The first term in $u_{ij}(t)$ is an empirical estimate of p_{ij} and the second term bounds the fluctuations of this estimate with high probability, see [1, 18]. The α parameter in the second term controls the width of the upper confidence bound.

Additionally, we maintain a second wider upper bound $v_{ij}(t)$, which we use to increase parallel exploration. We define $v_{ij}(t)$ by

$$v_{ij}(t) = \frac{w_{ij}(t)}{n_{ij}(t)} + \sqrt{\frac{\beta \alpha \ln t}{n_{ij}(t)}}, \qquad (3)$$

where the parameter $\beta \geq 1$ controls how much wider it is than the upper confidence bound of Equation 2. When there is more than one candidate for a Condorcet winner according to the "narrow" confidence bounds in Equation 2 an exploration round is triggered and arms that could be Condorcet winner candidates according to the "wide" confidence bounds are compared. This leads to some arms being explored preemptively and decreases the overall number of exploration rounds by increasing parallel exploration.

Given K arms, we define $U_i(t) = \min_{j \in K, j \neq i} \{u_{ij}(t)\}$, i.e. $U_i(t)$ is the smallest upper confidence bound of i. Let E denote the set of potential Condorcet winners, which contains all arms i for which $U_i(t) \geq 1/2$. Additionally, we define $V_i(t) = \min_{j \in K, j \neq i} \{v_{ij}(t)\}$ and F to be the set of potential Condorcet winners according to the wider upper bounds, that is, all arms for which $V_i(t) \geq 1/2$.

At each iteration of Algorithm 1, if there is only a single potential Condorcet winner in E, we choose this arm. If there are several potential Condorcet winners, we select all arms in the larger set F. In the unlikely event that there are no potential Condorcet winners, we select all arms.

4. EXPERIMENTAL EVALUATION

We next present the experimental evaluation of our Multi-Dueling Bandits (MDB) algorithm.

4.1 Experimental Setup

We begin by describing our experimental setup.

4.1.1 Baselines

We compare our MDB algorithm to three state-of-the-art dueling bandit algorithms, namely RUCB and MergeRUCB, both implemented in the freely available software package Lerot [11], and RMED1 [10]. As per [17], we set the α parameter of Equation 2 for RUCB to 0.51, and to 1.01 for MergeRUCB. For RMED1 we use the same parameter setting as [10]: $f(K) = 0.3K^{1.01}$. To select the parameters for MDB, we carried out a grid search on the grid $\{0.5, 1, 1.5\} \times \{1.25, 1.5, 2, 4\}$ on a separate dataset, specifically the validation set of the YLR1 dataset, and found the

---

1 $W = [w_{ij}] := 0_{K \times K}$
2 Play all arms and update the corresponding entries in W
3 **for** $t = 2, \ldots, T$ **do**
4 $\quad$ $U := [u_{ij}(t)] = \frac{w_{ij}(t)}{n_{ij}(t)} + \sqrt{\frac{\alpha \ln t}{n_{ij}(t)}}$, $u_{ii}(t) = 1/2$
5 $\quad$ $V := [v_{ij}(t)] = \frac{w_{ij}(t)}{n_{ij}(t)} + \sqrt{\frac{\beta \alpha \ln t}{n_{ij}(t)}}$, $v_{ii}(t) = 1/2$
6 $\quad$ $E = \{i$ s.t. $U_i(t) \geq 1/2\}$ (The set of potential champions according to U)
7 $\quad$ $F = \{i$ s.t. $V_i(t) \geq 1/2\}$ (The set of potential champions according to V)
8 $\quad$ **if** $|E| > 1$ **then**
9 $\quad\quad$ Choose all arms $f \in F$ for comparison and update the corresponding entries in W
10 $\quad$ **else if** $|E| = 1$ **then**
11 $\quad\quad$ Choose the arm $e \in E$
12 $\quad$ **else**
13 $\quad\quad$ Choose all arms for comparison and update the corresponding entries in W
14 **end**

Algorithm 1: Multi-Dueling Bandit (MDB) Algorithm.

Table 1: Datasets. Each dataset consists of a number of query-document pairs, together with a relevance judgement for the pair. Each document is represented by a feature vector.

Datasets	Queries	URLs	Features
MSLR-WEB30K [2]	31,531	3,771,125	136
YLR Set 1 [4]	19,944	473,134	700
YLR Set 2 [4]	1,266	34,815	700
Yandex [3]	9,124	97,290	245

best parameters to be $\alpha = 0.5$ and $\beta = 1.5$. We used these as our parameter settings for MDB for all other experiments.

4.1.2 Datasets

We compare the algorithms on four large-scale evaluation datasets summarised in Table 1[1]. Since there was no Condorcet winner for the Yandex dataset, we randomly sampled subsets of 200 rankers from the Yandex dataset, selecting the first subset with a Condorcet winner.

4.1.3 Ranker Construction

Following [17], for each dataset we choose the rankers to be the features of the dataset. That is, for a given feature, we construct a ranker which ranks documents only according to the score of that feature. As noted in [17] this is a somewhat artificial setup from a learning-to-rank perspective, however from the point of view of *evaluating* dueling and multi-dueling bandit algorithms, the difficulty of a problem instance is affected by the relative performance of the rankers, not their absolute performance. Using the feature rankers is therefore useful for assessing the performance of dueling and multi-dueling bandit algorithms since many of the features perform similarly and are therefore difficult to distinguish using interleaved or multileaved comparisons.

---

[1]Only 519 features are non-zero for YLR Set 1 and only 596 features are non-zero for YLR Set 2. The remaining features are zero for all query-document pairs.
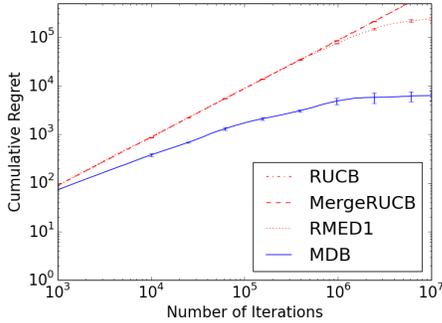
(a) YLR1

Figure 1: Cumulative regret averaged over 10 runs against number of iterations for the 4 algorithms on the YLR1 dataset with the navigational click model.

4.1.4 Simulated User Model

All experiments are conducted using a simulated user model. For each iteration we randomly sample with replacement one query from the pool of queries of the dataset. The dueling or multi-dueling bandit algorithms choose rankers, whose results are then interleaved or multileaved respectively, and presented to a simulated user. For the dueling bandit algorithms, we compare pairs of rankers using probabilistic interleaving [8]. For MDB, we use SOSM [2]. Both probabilistic interleaving and SOSM only present the top-10 documents to users. Clicks are then generated from a probabilistic user model [9]. The interleaving or multileaving algorithm scores the chosen rankers using the clicks generated by the user model.

The click model used for these experiments was the navigational user model from [9], unless otherwise stated. This click model has been used as a standard click model for dueling bandit algorithm evaluation in [17].

4.2 Results

Below we summarize the experimental results for the various experiments. For all figures, the error bars show the standard deviation of cumulative regret across runs for each algorithm at the given time step.

4.2.1 Experiments on Real Learning-to-Rank Datasets

Figure 1 shows how the cumulative regret increases with each iteration using the real learning to rank dataset YLR1. It shows that MDB substantially outperforms the baselines. It outperforms the best dueling bandit algorithm, RMED1, by almost 2 orders of magnitude. RMED1 outperforms RUCB and MergeRUCB, as expected from the results of [10]. The results for the other datasets are similar, and can be seen in the extended version [3], with MDB outperforming the best dueling bandit algorithm, RMED1, by a factor of 3 in the dataset with the smallest number of features, and orders of magnitude for the other datasets.

Note that for the Yandex dataset, since there was no Condorcet winner, we randomly sampled 200 of the 245 feature rankers to obtain a dataset with a Condorcet winner. Results using the full Yandex dataset with no Condorcet winner are described in the extended version [3].

4.2.2 Dependence on Number of Rankers

To isolate the impact of the number of rankers being com-
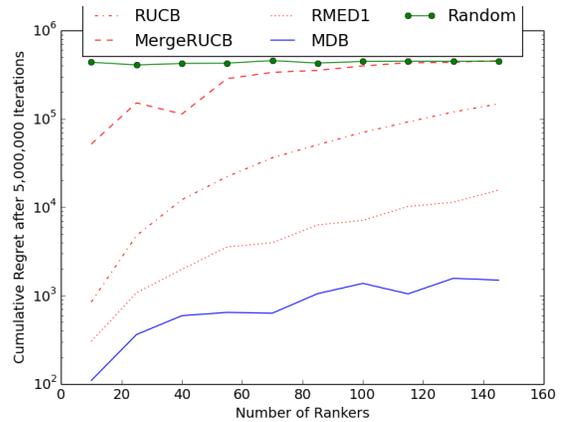
Figure 2: Cumulative regret averaged over 10 runs after 5,000,000 iterations against number of rankers for the 4 algorithms, and a random policy, on subsets with M rankers of the YLR1 dataset with navigational click model.

pared on the real datasets involving multileaving, we investigate how regret scales with the number of rankers being compared using the YLR1 dataset. We randomly sampled subsets of rankers of sizes $\{10, 25, 40, 55, 70, 85, 100, 115, 130, 145\}$ from the YLR1 dataset. Note that we randomly sampled different subsets of rankers for each run. For each of these subsets we then carried out 10 runs of each algorithm over 5,000,000 iterations and recorded the average cumulative regret across runs.

Figure 2 shows how the performance of the 4 algorithms varies as a function of the number of rankers. Additionally we have shown the performance of a random policy which simply selects a random subset of the rankers for multileaving at each iteration. We observe that as the number of rankers increases the cumulative regret increases most for RUCB and MergeRUCB, while it increases more slowly for RMED1. Regret appears to be almost independent of the number of rankers for MDB.

These experiments were also carried out for the perfect and informational click models, and the results were very similar but have been omitted due to page restrictions.

For the MDB algorithm, the regret associated with having to explore suboptimal rankers does not appear to be additionally compounded by the number of rankers being explored. This is an important characteristic of the MDB algorithm, since if we can explore additional rankers with no substantial additional cost, the risks associated with large-scale online ranker evaluation are substantially mitigated.

Note that it may appear that regret levels off for MergeRUCB as we increase the number of rankers. This is due to the fact that for 5,000,000 iterations there is a limit to how much regret can be incurred just by making random choices in 5,000,000 iterations. For larger problem sizes and for a time frame of 5,000,000 iterations, MergeRUCB begins to perform no better than a random policy. This does not imply that MergeRUCB performs as badly as a random policy in general, but for these problem instances it has not yet begun to eliminate suboptimal arms after 5,000,000 iterations. Further iterations would be needed to show improvements relative to the random policy.
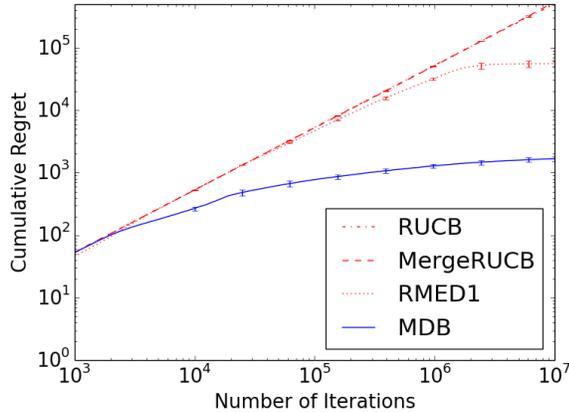
(a) informational click model

Figure 3: Cumulative regret averaged over 10 runs against number of iterations for the 4 algorithms on the YLR1 dataset using the informational click model.

4.2.3 Dependence on Click Model

To test the robustness of our approach to the choice of click model, we also investigated performance using the perfect and informational click models [9]. Figure 3 shows the cumulative regret for the informational click model, using randomly selected subsets of size 200 of the rankers from the YLR1 dataset. We chose to use subsets of the full dataset for these experiments because of the computational costs of running RMED1 on the full YLR1 dataset. For all click models MDB outperforms the best dueling bandit algorithm by between 1 and 2 orders of magnitude, and the results for the remaining click models are included in the extended version [3].

For MDB, the regret doubles when going from the perfect to the navigational click model, but does not increase further for the informational click model. In contrast, for the dueling bandit algorithms, regret for the informational click model is approximately double that for the navigational click model, which is approximately double that of the perfect click model. MDB is therefore least affected by varying the click model in our experiments.

4.2.4 Distortion of probability estimates due to multileaving

As discussed earlier, simultaneous comparison of more than two arms may affect their pairwise winning probabilities. We called this effect *distortion*. We can quantify this effect by first randomly sampling a fixed size subset of rankers that includes a Condorcet winner, and then measuring, after some fixed number of multileavings, the fraction of rankers that beat the Condorcet winner more than 50% of the time. If there is no distortion, and the number of multileavings is sufficient, we expect this fraction to be zero.

In these experiments we test the level of distortion in the multileaving method SOSM, and examine how robust our MDB algorithm is to possible distortions in the multileaving method.

For each dataset, and each click model, we randomly sample subsets of rankers of sizes 3, 10, and 100 that include a Condorcet winner. We examine the probabilities of the

rankers beating the Condorcet winner after 3,000 multileavings. Note that this is likely to be an overestimate of the distortion of the multileaving method, since, for rankers of very similar quality, 3,000 iterations may not be sufficient to reliably distinguish rankers. Table 2 shows the average, over 30 runs, of the percentage of rankers that beat the Condorcet winner. Note that we have only included the results for the informational click model, the results for the remaining click models are shown in the extended version [3].

We observe that the distortion problem is almost unique to the MSLR dataset, and is exacerbated by the noisiest click model, the informational click model. The distortion problem is exclusively related to the feature ranker 133 in the MSLR dataset. Feature ranker 133 scores documents solely based on the query-document clicks, i.e. a document was clicked on in response to a query. This feature is very good at identifying 1 or 2 documents that are very likely to be relevant. However, when asked to rank documents in a multileaved set, most of the documents, even though they might be relevant, have not been previously clicked on. As such, ranker 133 is unable to distinguish between the vast majority of documents. Thus, even though ranker 133 performs well in pairwise comparisons, where it has contributed half of the documents in the results list, it performs very poorly when multileaved with many other rankers. Table 2 also includes results for the MSLR dataset, when feature ranker 133 is excluded. This is denoted by MSLR*. When ranker 133 is excluded, no substantial distortion is observed.

The moderate levels of distortion observed for the Yandex and MSLR dataset (excluding feature ranker 133) are likely to be mostly caused by the fact that there are many rankers that are very similar in quality, and so 3,000 comparisons are not sufficient to differentiate these similar rankers.

Table 2: Percentage of rankers beating the Condorcet winner (distortion), averaged over 30 runs, after 3,000 iterations for 3, 10, and 100 rankers being multileaved for each dataset and click model. The dataset denoted MSLR* is the MSLR dataset with feature ranker 133 removed.

	Distortion		
Num. Rankers	3	10	100
MSLR Informational	0.0%	6.8%	41.3%
MSLR* Informational	0.0%	2.9%	2.7%
YLR1 Informational	0.0%	0.0%	0.3%
YLR2 Informational	0.0%	1.0%	0.9%
Yandex Informational	3.3%	3.9%	3.4%

The only problem setting where our MDB algorithm did not substantially outperform the best baseline dueling bandit algorithm, RMED1, was for the MSLR dataset with all 136 feature rankers with the informational click model. The results for this problem setting are shown in Figure 4. This is due specifically to the feature ranker 133 in the MSLR dataset. In the extended version [3], we show that there was some distortion for all click models for the MSLR dataset. However, it is with the informational click model that the distortion is greatest, reaching 41.3% for 100 rankers. This is a very high percentage. The MDB algorithm appears to be robust to more reasonable levels of distortion, suffering substantially less regret than the baselines for the MSLR
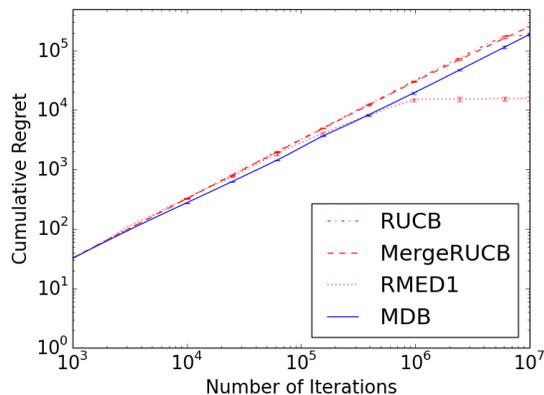
Figure 4: Average cumulative regret over 10 runs against number of iterations for the 4 algorithms on the MSLR dataset using the informational click model.

dataset with the navigational click model, and with the perfect click model (this result is omitted due to the page restriction). Additionally, for the MSLR dataset with feature ranker 133 removed, MDB substantially outperformed all baselines for all click models. Results are omitted due to the page restriction.

5. CONCLUSIONS AND FUTURE WORK

This paper proposes a generalisation of the K-armed dueling bandits termed multi-dueling bandits (MDB). We have applied MDB in online ranker evaluation to leverage the power of simultaneous comparisons through multileaving and improve the exploration-exploitation trade-off. Our experimental results on data from 4 standard datasets demonstrated up to 1 to 2 orders of magnitude reduction in regret compared to state-of-the-art dueling bandit algorithms, RUCB [18], MergeRUCB [17], and RMED1 [10] in all except one pathological case discussed below. Generally, relative benefits compared to dueling bandits increased with the number of rankers being compared. For MDB, the incurred regret did not increase substantially as the number of rankers increased. As such, the risks associated with large-scale online ranker evaluation are substantially mitigated. Further experiments showed that MDB was robust to various user click models.

Experiments were also conducted to examine the behaviour of MDB in the absence of a Condorcet winner, which is the case for the full Yandex dataset. In this case, the regret was approximated by measuring the NDGC@10 score. In this case MDB outperformed the best dueling bandit algorithm, RMED1, by approximately an order of magnitude after 5,000,000 iterations.

We also investigated the level of distortion of pairwise winning probabilities in multileaving using SOSM. For the MSLR dataset using a navigational click model, the distortion reached 41.3%. In this case MDB was inferior to RMED1. The high level of distortion was due to the peculiarities of ranker 133. If ranker 133 is removed, the distortion of pairwise winning probabilities is significantly reduced and MDB outperforms all other algorithms.

There are a number of avenues for future work. The distortion of pairwise winning probabilities in multileaving

needs further investigation. All existing multileaving algorithms exhibit this behaviour. It remains an open question as to whether a new multileaving algorithm can be designed to avoid this problem, or at least minimize it. Furthermore, a theoretical analysis of our algorithm needs to be developed to better understand its power and limitations. Additionally, since a Condorcet winner is not guaranteed to exist, it may be useful to explore other concepts of winners, such as the Copeland [16], Borda [14] and von Neumann [6] criteria. Finally, we note that the proposed multi-dueling bandit algorithm can be applied to a broad class of problems and applications in other domains, e.g. recommender systems.

6. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[2] B. Brost, C. Lioma, Y. Seldin, and I. J. Cox. An improved multileaving algorithm for online ranker evaluation - in press. SIGIR, 2016.

[3] B. Brost, Y. Seldin, I. J. Cox, and C. Lioma. Multi-dueling bandits and their application to online ranker evaluation. *arXiv preprint arXiv:1506.00312*, 2016.

[4] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. In *Yahoo! Learning to Rank Challenge*, pages 1–24, 2011.

[5] O. Chapelle, T. Joachims, F. Radlinski, and Y. Yue. Large-scale validation and analysis of interleaved search evaluation. *TOIS*, 30(1):6, 2012.

[6] M. Dudík, K. Hofmann, R. E. Schapire, A. Slivkins, and M. Zoghi. Contextual dueling bandits. *arXiv preprint arXiv:1502.06362*, 2015.

[7] K. Hofmann, S. Whiteson, and M. de Rijke. Balancing exploration and exploitation in learning to rank online. In *Advances in Information Retrieval*, pages 251–263. Springer, 2011.

[8] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM*, pages 249–258. ACM, 2011.

[9] K. Hofmann, S. Whiteson, and M. D. Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *TOIS*, 31(4):17, 2013.

[10] J. Komiyama, J. Honda, H. Kashima, and H. Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. *arXiv preprint arXiv:1506.02550*, 2015.

[11] A. Schuth, K. Hofmann, S. Whiteson, and M. de Rijke. Lerot: An online learning to rank framework. In *Proceedings of the 2013 workshop on Living labs for information retrieval evaluation*, pages 23–26. ACM, 2013.

[12] A. Schuth, F. Sietsma, S. Whiteson, D. Lefortier, and M. de Rijke. Multileaved comparisons for fast online evaluation. In *CIKM*, pages 71–80. ACM, 2014.

[13] A. Schuth et al. Probabilistic multileave for online retrieval evaluation. SIGIR, 2015.

[14] T. Urvoy, F. Clerot, R. Féraud, and S. Naamane. Generic exploration and k-armed voting bandits. In *ICML*, pages 91–99, 2013.

[15] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[16] M. Zoghi, Z. Karnin, S. Whiteson, and M. de Rijke. Copeland dueling bandits. *arXiv preprint arXiv:1506.00312*, 2015.

[17] M. Zoghi, S. Whiteson, and M. de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *WSDM*, 2015.

[18] M. Zoghi, S. Whiteson, R. Munos, M. d. Rijke, et al. Relative upper confidence bound for the k-armed dueling bandit problem. In *JMLR Workshop and Conference Proceedings*, number 32, pages 10–18. JMLR, 2014.

[19] M. Zoghi, S. A. Whiteson, M. de Rijke, and R. Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *WSDM*, pages 73–82. ACM, 2014.