Advancements in Dueling Bandits

Yanan Sui[1], **Masrour Zoghi**[2], **Katja Hofmann**[2], **Yisong Yue**[1]
[1] Caltech
[2] Microsoft
ysui@caltech.edu, masrour@zoghi.org, katja.hofmann@microsoft.com, yyue@caltech.edu

Abstract

The dueling bandits problem is an online learning framework where learning happens "on-the-fly" through preference feedback, i.e., from comparisons between a pair of actions. Unlike conventional online learning settings that require absolute feedback for each action, the dueling bandits framework assumes only the presence of (noisy) binary feedback about the relative quality of each pair of actions. The dueling bandits problem is well-suited for modeling settings that elicit subjective or implicit human feedback, which is typically more reliable in preference form. In this survey, we review recent results in the theories, algorithms, and applications of the dueling bandits problem. As an emerging domain, the theories and algorithms of dueling bandits have been intensively studied during the past few years. We provide an overview of recent advancements, including algorithmic advances and applications. We discuss extensions to standard problem formulation and novel application areas, highlighting key open research questions in our discussion.

1 Introduction

Many real world applications require algorithms to simultaneously predict actions and learn "on-the-fly". Examples include implicit or subjective feedback for information retrieval and recommender systems [Chapelle *et al.*, 2012], personalized clinical treatments [Sui and Burdick, 2014], and many others. In these settings, the recommendation algorithm does not know a priori which actions are most effective, and must learn from trial and error. A grand technical question in this space is how to design algorithms that can quickly converge to recommending the optimal actions. Furthermore, in many settings, especially the ones that involve human feedback, it is more practical to elicit pairwise preferences, which are often more reliable than absolute feedback [Chapelle *et al.*, 2012].

Online learning is the setting where the learner is simultaneously acting (or predicting) and learning "on-the-fly". The goal is to be competitive w.r.t. some benchmark. One common benchmark is being competitive with the best fixed action with the benefit of hindsight. The difference between rewards accumulated by the best hindsight action and the actions of the learner is often called the cumulative regret, and hence a commonly studied version of online learning is online regret minimization.

In the bandits setting, also known as "partial-information" online learning, only the reward of the chosen action is revealed. Regret minimization in the bandit setting typically leads to an exploration-exploitation trade-off. On the one hand, it is important to select actions with (estimated) high reward. However, actions that appear very good may be sub-optimal due to imperfections in the learner's knowledge. Therefore, it is important to explore by performing seemingly bad actions in order to collect more information about them.

While the problem of online regret minimization is well-studied given cardinal feedback, it is less clear how to formulate regret when one only receives preference feedback. Given a set of K arms, we want to find a sequence of noisy comparisons to minimize cumulative regret. The dueling bandits problem provides one such formulation (see Equation (2) in Section 2.2), and defines regret of the chosen actions using the preference regret relative to the optimal action (e.g., how much one would have preferred the optimal action versus the chosen ones). Note that such a formulation assumes a so-called *Condorcet winner*, where there is a unique optimal action superior to all other actions. The Condorcet concept is the most studied one for dueling bandits. Other concepts of winners also exist but are less common.

In this survey, we overview recent advances in research on dueling bandits. For a thorough review of early work on dueling bandits, we refer readers to [Busa-Fekete and Hüllermeier, 2014]. The remainder of this paper is structured as follows. We overview the key concepts for problem setup in Section 2. Methods for the original dueling bandits problem are introduced in Section 3. Section 4 continues to discuss a variety of generalizations and applications of the dueling bandits problem. Section 5 concludes the survey.

2 Problem Setup

We first overview the definitions of regret for the multi-armed (Section 2.1) and dueling (Section 2.2) bandit settings, and follow with presenting various dueling bandits algorithms. For different algorithms, there is a range of modeling assumptions which lead to theoretical and/or empirical behaviors.

2.1 Multi-armed Bandits (MAB)

We provide a brief formal description of the conventional MAB problem for completeness. The stochastic MAB problem [Robbins, 1952] is an iterative decision making problem where the algorithm repeatedly chooses among K actions (or bandits or arms). The learner receives an absolute reward that depends on the action selected. We assume w.l.o.g. that every reward is bounded between $[0, 1]$ and is sampled independently with expected value $\mu(a)$ for each action a. The goal then is to minimize the cumulative regret compared to the best arm in expectation:

$$R_T^{\text{MAB}} = \sum_{t=1}^{T} \left[\mu^* - \mu(a(t))\right], \tag{1}$$

where $a(t)$ denotes the arm chosen at time t, $\mu(a)$ denotes the expected reward of arm a, and $\mu^* = \arg\max_a \mu(a)$.

In the adversarial setting, the rewards are chosen in an adversarial fashion, rather than sampled independently from some underlying distribution. In this case, regret in Equation (1) is rephrased as the difference in the sum of rewards.

Exploration-Exploitation Tradeoff. The issue of exploration versus exploitation becomes apparent when considering the consequences of minimizing Equation (1). Since the true expected rewards are unknown, one can only maintain an estimate from the observations of performed actions. Choosing bad actions will lead to high regret in Equation (1), but the algorithm can only determine an action is suboptimal from trying them. However, over-exploring can lead to slow convergence, which is also undesirable. Indeed, properly balancing the exploration-exploitation tradeoff is a central question in the study of sequential decision making under uncertainty.

2.2 Dueling Bandits

In the dueling bandits problem, the following happens for each time step $t = 1, \ldots, T$:

- The algorithm chooses a pair of actions a_i, a_j from K available actions.
- The world provides (independent stochastic) preference feedback of which action is more preferred. The first action is preferred with probability $P(a_i \succ a_j)$, and the second with probability $P(a_i \succ a_j) = 1 - P(a_j \succ a_i)$.

Unlike the standard multi-armed bandits setting described in Section 2.1, the dueling bandits problem requires choosing two arms, a_i and a_j. Furthermore, the feedback is either a_i or a_j as the winner of the comparison between the two arms (rather than an absolute reward). These preference probabilities form the entries of a $K \times K$ *preference matrix* $\mathbf{P}$, which defines the hidden information in the dueling bandits problem and is not revealed to the algorithm.

Table 1 illustrates an example 6×6 preference matrix. Here, the goal is to optimize over the set of six arms $\{A, B, C, D, E, F\}$. At each iteration, the agent picks two arms (arm B and arm E in this example) and compare them. The value 0.08 high-lighted in yellow shows that arm B has its winning probability of $(0.5 + 0.08)$ against arm E. Arm A is the Condorcet winner of the six arms as it beats any other arm with probability greater than 0.5.

In a similar fashion to the MAB setting, we wish to define a notion of regret to benchmark the performance of dueling bandits algorithms. However, the definition of regret is less clear-cut in this setting, due to the nature of preference feedback instead of cardinal feedback. One common approach is to define regret relative to the a best arm, under a suitable definition of "best". The most straightforward case is a Condorcet winner, where one arm is preferred to all other arms. We discuss different solution concepts later in the paper.

To simplify the notation in the rest of the paper, we relabel the arms such that a_1 the best arm (e.g., the Condorcet winner). The cumulative regret after T time-steps is:

$$\mathcal{R}(T) = \sum_{t=1}^{T} r(t), \qquad r = \Delta_{1i} + \Delta_{1j}. \tag{2}$$

where the *instantaneous regret* $r(t)$ is the regret incurred by the choice of arms at time t, and $\Delta_{1k} := P_{1k} - 0.5 = P(a_1 \succ a_k) - 0.5$ for each k.

Comparison to Multi-armed Bandits. One major challenge of the dueling bandits problem stems from the fact that the algorithm cannot directly observe the costs of the chosen actions. It is an example of a partial monitoring problem, a class of regret-minimization problems defined in [Cesa-Bianchi *et al.*, 2006], in which the algorithm observes feedback that depend on the actions chosen by the forecaster and by an unseen opponent (the "environment"). This pair of actions also determines a loss, which is not revealed to the learner but is used in defining the regret. For instance, in Equation (2), regret is measured relative to the unknown best action, but the learner only observes feedback involving the two selected actions. As an example, consider Table 1, which depicts a stochastic preference matrix over six actions, with action A being the best. If the two selected actions are B and E, then the learner suffers regret $(\Delta_{AB} + \Delta_{AE})$ despite collecting feedback to reveals information about Δ_{BE}.

2.3 Different Solution Concepts

There are a variety of ways one can define a solution concept and thus regret. As mentioned above, the most straightforward case is where there is a Condorcet winner which is preferred to all other arms, i.e., an arm a_C such that $P_{Cj} > 0.5$ for all $j \neq C$. However, Table 2 shows a preference matrix which doesn't not have a Condorcet winner ($A \succ B$, $B \succ C$, and $C \succ A$ in this example).

		P2					
		A	B	C	D	E	F
P1	A	0	0.03	0.04	0.06	0.10	0.11
	B	-0.03	0	0.03	0.05	**0.08**	0.11
	C	-0.04	-0.03	0	0.04	0.07	0.09
	D	-0.06	-0.05	-0.04	0	0.05	0.07
	E	-0.10	-0.08	-0.07	-0.05	0	0.03
	F	-0.11	-0.11	-0.09	-0.07	-0.03	0

Table 1: Preference matrix of 6 arms, sorted in preference order. The first choice (P1) is B, and the second (P2) is E. The probability of P1 defeating P2 is $(0.08 + 0.5)$. Incurred regret is $\Delta_{AB} + \Delta_{AE}$.

	A	B	C	D	E	F
A	0	**0.03**	**-0.02**	0.06	0.10	0.11
B	-0.03	0	**0.03**	0.05	0.08	0.11
C	0.02	-0.03	0	0.04	0.07	0.09
D	-0.06	-0.05	-0.04	0	0.05	0.07
E	-0.10	-0.08	-0.07	-0.05	0	0.03
F	-0.11	-0.11	-0.09	-0.07	-0.03	0

Table 2: Violation of Condorcet Winner. Highlighted entries are different from Table 1. No Condorcet winner exists as no arm could beat every other arm.

	A	B	C	D	E	F
A	0	**0.03**	**0.02**	0.06	0.10	0.11
B	-0.03	0	**0.03**	0.05	0.08	0.11
C	-0.02	-0.03	0	0.04	0.07	0.09
D	-0.06	-0.05	-0.04	0	0.05	0.07
E	-0.10	-0.08	-0.07	-0.05	0	0.03
F	-0.11	-0.11	-0.09	-0.07	-0.03	0

Table 3: Violation of Transitivity. Highlighted entries are different from Table 1. $A \succ B \succ C$, but $\Delta_{AC} < \Delta_{BC}$.

There are numerous proposals in the literature for alternative notions of winners in the absence of a Condorcet winner, e.g., Borda winner [Urvoy *et al.*, 2013; Jamieson *et al.*, 2015], Copeland winner [Zoghi *et al.*, 2015a; Komiyama *et al.*, 2016], von Neumann winner [Dudík *et al.*, 2015], with each definition having its own benefits and drawbacks. We will discuss some of these extensions in Section 4.

3 Dueling Bandits Algorithms with Condorcet Winners

In this section, we review dueling bandits algorithms in the literature that solve the Condorcet dueling bandits problem, i.e. problems where there exists a single arm that is preferred to all other arms. There are basically two styles of algorithm design: Asymmetric Algorithms and Symmetric Algorithms. The first style (pseudocode shown in Algorithm 1) conceptually separates the two choices into choosing a reference arm and a exploration arm. The reference arm is typically chosen either due it being the best known action thus far, or as one of the plausibly best actions (i.e., not yet eliminated as possibly being the best action). This includes the algorithms of IF, BtM, SAVAGE, Doubler, RUCB, MergeRUCB, RCS, and DTS. The exploration arm is then chosen to duel against the reference arm, e.g., to identify an arm that can outperform the reference arm.

The second style (pseudocode shown in Algorithm 2) treats the choise of the two arms symmetrically. For instance, one simple approach is to use a separate online learning algorithm to choose each arm, which shares affinity to online learning in repeated zero-sum games [Cesa-Bianchi and Lugosi, 2006]. The intuition is that both online learners should converge to the global optimum (Condorcet winner), which corresponds to the unique pure Nash equilibrium when viewing the preference matrix as a zero-sum two-player payoff matrix. This style of algorithms includes Sparring & Self-Sparring.

Algorithm 1 Asymmetric Algorithmic Framework

1: **for** $t = 1, 2, \ldots$ **do**
2: Choose a reference arm a_i.
3: Given the arm a_i, choose an exploratory arm a_j to explore against the reference arm.
4: Duel the chosen arms (a_i, a_j), observe feedback: $a_i \succ a_j$ or $a_j \succ a_i$.
5: Integrate feedback to update corresponding arms.
6: **end for**

Algorithm 2 Symmetric Algorithm Framework

1: **for** $t = 1, 2, \ldots$ **do**
2: Choose arm a_i and arm a_j symmetrically.
3: Duel the chosen arms (a_i, a_j), observe feedback: $a_i \succ a_j$ or $a_j \succ a_i$.
4: Integrate feedback to update corresponding arms.
5: **end for**

IF and BtM

The first two methods proposed for the K-armed dueling bandits problem are Interleaved Filter (IF) [Yue *et al.*, 2012] (conference version published in 2009) and Beat the Mean (BtM) [Yue and Joachims, 2011]. These methods work under the following assumptions:

1. A total ordering of the arms, i.e. we can relabel the arms as $a_1, \ldots, a_K$ such that $p_{ij} > 0.5$ for all $i < j$. This assumption implies a Condorcet winner.

2. Stochastic Triangle Inequality (STI): for any pair (j, k), with $1 < j < k$, the following condition is satisfied: $\Delta_{1k} \leq \Delta_{1j} + \Delta_{jk}$, where $\Delta_{ij} := p_{ij} - 0.5$.

3. IF and BtM require two different transitivity conditions:

IF: Strong Stochastic Transitivity (SST): for any triple (i, j, k), with $i < j < k$, the following condition is satisfied: $\Delta_{ik} \geq \max\{\Delta_{ij}, \Delta_{jk}\}$.

BtM: Relaxed Stochastic Transitivity (RST): there exists a number $\gamma \geq 1$ such that for all pairs (j, k) with $1 < j < k$, we have $\gamma\Delta_{1k} \geq \max\{\Delta_{1j}, \Delta_{jk}\}$.

Table 3 shows a case where Strong Stochastic Transitivity does not hold. In BtM, the constant γ, which measures the degree to which SST fails to hold, needs to be passed to the algorithm explicitly: the higher the γ, the more challenging the problem, with SST holding when $\gamma = 1$. Given these assumptions, the following regret bounds have been proven for IF and BtM. For large T, we have:

$$\mathbb{E}\left[R_T^{\mathrm{IF}_T}\right] \leq \mathcal{O}\left(\frac{K \log T}{\Delta_{\min}}\right), \text{ and}$$

$$R_T^{\mathrm{BtM}_T} \leq \mathcal{O}\left(\frac{\gamma^7 K \log T}{\Delta_{\min}}\right) \text{ with high probability,}$$

where R_T is cumulative regret in the Condorcet setting, defined in §2.2. Moreover, $\Delta_{\min}$ is the smallest gap $\Delta_{1j} := p_{1j} - 0.5$, where a_1 is the best arm.

The first bound holds only when $\gamma = 1$ but matches the lower bound in [Yue *et al.*, 2012, Theorem 2]. The second bound holds for $\gamma \geq 1$ and is sharp when $\gamma = 1$.

IF is based on a form of "hill climbing." IF begins by choosing a random arm $\widehat{a}$ as the reference arm and compares

it against the other arms until we realize (with high probability) that $\widehat{a}$ loses to another arm, at which point the algorithm pivots to the latter arm as the reference arm. Afterwards, IF restarts the process with the new reference arm. Additionally, the algorithm keeps track of the arms that are beaten by any reference arm $\widehat{a}$ and eliminates them from consideration, reducing the need to explore them against future reference arms. By exploiting stochastic triangle inequality and strong stochastic transitivity, one can show fast improvement of the sequence of reference arms towards the Condorcet winner (in a logarithmic number of rounds), as well as exponentially fast elimination of suboptimal arms in expectation (a constant fraction of arms eliminated against each $\widehat{a}$), thus leading to the regret guarantee.

To better understand BtM, we begin by first defining the following quantity: given a $K \times K$ preference matrix $\mathbf{P} = [p_{ij}]$, define the *Borda score* of arm a_i as $\frac{1}{K}\sum_j p_{ij}$. The key observations behind BtM are the following:

1. First, the Borda score of the Condorcet winner is always greater than or equal to 0.5 because by definition the Condorcet winner beats all other arms with probability greater than 0.5. Therefore, the Condorcet winner is not a "Borda loser" and as long as we eliminate Borda losers, the Condorcet winner would not be eliminated.

2. Second, the other important property of the Condorcet winner of a dueling bandits problem is that it remains the Condorcet winner of any dueling bandits problem obtained by removing any arm other than the Condorcet winner.

Putting these two observations together, we see that as long as we keep eliminating Borda losers, we will eventually be left with nothing but the Condorcet winner. Compared to IF, BtM offers more stable performance (a high probability regret bound rather than in expectation) due to the variability in how quickly IF eliminates arms against the chosen reference arms.

SAVAGE

Sensitivity Analysis of VAriables for Generic Exploration (SAVAGE) [Urvoy *et al.*, 2013] is an algorithm that empirically outperforms both IF and BtM by a wide margin when the number of arms is of moderate size, as demonstrated by the experimental results in [Urvoy *et al.*, 2013]. One version of SAVAGE, called *Condorcet SAVAGE*, makes the Condorcet assumption and has a regret bound in the form $\mathcal{O}(K^2 \log T)$, which is not as tight as those of IF and BtM.

At high level, the algorithm compares pairs of arms in a round robin fashion and drop pairs of arms from consideration as soon as it is safe to do so, according to the following rule. If we know that the dueling bandits problem has a Condorcet winner, then any arm that loses with high probability to another arm cannot be a Condorcet winner and so can be eliminated from further consideration. Proceeding in this fashion, we will eventually be left with nothing but the Condorcet winner, which is precisely how Condorcet SAVAGE finds the Condorcet winner.

Doubler

Doubler [Ailon *et al.*, 2014], is the first approach which converts dueling bandits into conventional multi-armed bandit problems, under the assumption that the preferences are linear choice functions of underlying utilities associated with the arms. In other words, $\Delta_{AB} = (\mu_A - \mu_B)/2$.

Doubler proceeds in epochs of exponentially increasing size (hence "doubler"). In each epoch, the left arm is sampled from a fixed distribution, and the right arm is chosen using a multi-armed bandit algorithm to minimize regret against the left arm. The feedback received by the multi-armed bandit algorithm is the wins and losses the right arm encounters when compared against the left arm. In other words, the goal of the right arm is to beat the fixed distribution from which the left arm is sampled. The distribution the left arm plays is the empirical distribution (histogram) of arms that were chosen for the right arm in the previous epoch.

Since the utility assumption induces a total ordering, the algorithm provably converges to the best arms. While the regret bounds are near-optimal up to constant factors, in the practice Doubler is not efficient compared to other algorithms due to the doubling trick being conservative in how long it takes to switch to a new distribution of left arms. Furthermore, the linearity assumption may be overly restrictive in practice.

RUCB, MergeRUCB, RCS and DTS

Relative Upper Confidence Bounds (RUCB) [Zoghi *et al.*, 2014b] extends UCB to dueling bandits using a matrix of optimistic estimates of the preference probabilities. The RUCB algorithm significantly improves both theoretical and experimental results of dueling bandits. For instance, RUCB achieves high-probability regret bounds while making minimal assumptions other than assuming a Condorcet winner. The experimental success of RUCB over its predecessors is largely due to the fact that it avoids arm elimination.

At each time-step, RUCB chooses the first arm to be one that beats all other arms given an optimism bonus in its favor (i.e. a contender for the Condorcet winner), and chooses the second arm to be the arm that beats the first arm given an optimism bonus in the favor of the former, which translates to a pessimism penalty for the first arm. In particular, for an arm to be compared against itself, it needs to beat all other arms both optimistically and pessimistically. Indeed, one of the shortcomings of RUCB in practice is that it is overly prudent when it comes to comparing the Condorcet winner against itself and so it continues to accumulate regret for a long time.

The cumulative regret of RUCB after T time-steps is bounded by an expression of the form $\mathcal{O}(K^2 + K \log T)$, which improves upon the regret bound for Condorcet SAVAGE, but continues having a quadratic dependence on the number of arms, K, which poses a problem when dealing with large-scale problems. This issue was resolved by the introduction of MergeRUCB [Zoghi *et al.*, 2015b], which does away with the K^2 term using a divide-and-conquer strategy.

More specifically, MergeRUCB partitions the K arms into small batches and compares arms within each batch. An arm is eliminated from a batch once we realize that even according to the most optimistic estimate of the preference probabilities it loses to another arm in the batch. Once enough arms have been eliminated, MergeRUCB repartitions the arms and continues as before. Importantly, MergeRUCB does not require global pairwise comparisons between all pairs of arms, and

so its cumulative regret can be bounded by $O(K \log T)$.

To address the other shortcoming of RUCB mentioned above, namely its hesitance to begin exploiting before it is certain that it has found the correct Condorcet winner, one can employ an approach based upon Thompson Sampling, rather than confidence bounds. This is what was partially carried out in the case of Relative Confidence Sampling (RCS) [Zoghi *et al.*, 2014a]. More specifically, the contender for the Condorcet winner is chosen using samples from Beta posteriors on the preference probabilities.

RCS was shown to perform better than RUCB experimentally, but it lacks theoretical guarantees. However, a similar approach was proposed under the name of Double Thompson Sampling (DTS) [Wu and Liu, 2016], which does come equipped with regret bounds. DTS improves upon RUCB by using Thompson Sampling to break ties when choosing the first arm in RUCB and uses another round of Thompson Sampling to choose the second arm. The cumulative regret of DTS is bounded by $O(K \log T + K^2 \log \log T)$.

DTS is the state-of-the-art in the case of small-scale dueling bandits problems, while MergeRUCB is the state-of-the-art for large-scale dueling bandits algorithms. In fact, as discussed in the next section, DTS solves the more general Copeland dueling bandits problem, while RUCB, RCS and MergeRUCB are limited to the Condorcet setting.

RMED

The Relative Minimum Empirical Divergence (RMED) algorithm has been proposed by [Komiyama *et al.*, 2015] as an algorithm with an optimal asymptotic regret bound, which improves upon the results for RUCB, since the regret bound for RUCB does not match the lower bound proven in [Komiyama *et al.*, 2015]. The authors prove a lower bound on the cumulative regret of any dueling bandits algorithm, which takes the form

$$R_T \geq \sum_{k=2}^{K} \min_{\{j|p_{ij}<.5\}} \frac{(\Delta_{1i} + \Delta_{1j}) \log T}{2d(p_{ij}, .5)},$$

where $d(p, q) := p \log \frac{p}{q} + (1-q) \log \frac{1-p}{1-q}$. This lower bound follows from an intermediate result showing that for any dueling bandit algorithm with sublinear regret and any suboptimal arm $i > 1$, the number of comparisons between i and any arm j that beats i in the first T time-steps (denoted by $N_i j(T)$) can be bounded from below as follows:

$$\mathbb{E}\left[\sum_{\{j|p_{ij}<.5\}} d(p_{ij,}.5) N_{ij}(T)\right] \geq c \log T$$

for some positive constant c.

The upper bound for RMED matches this lower bound asymptotically. Indeed, the algorithm is directly inspired by the lower bound, in the sense that the algorithm maintains empirical estimate of the sum of divergences mentioned above. More specifically, the algorithm computes for each arm the empirical divergence defined as follows:

$$I_i(t) := \sum_{\{j|p_{ij}<.5\}} d(\widehat{p}_{ij}(t), .5) N_{ij}(t),$$

where $\widehat{p}_{ij}(t)$ is the algorithm's empirical estimate of the preference probability p_{ij} at time t. The authors consider $\exp(-I_i(t))$ to be the likelihood that arm i is the Condorcet winner and use this to pick the arms that are going to be compared against each other at time $t+1$. Despite its asymptotic optimality, the regret bound for RMED has a quadratic dependence on the number of arms.

Sparring & Self-Sparring

We now introduce the second trend of algorithm design which treats the two dueling arms symmetrically, such as a game played by two agents.

Sparring [Ailon *et al.*, 2014] is an elegant method for converting dueling bandit problems into MAB problems, typically under the linear choice model as assumed in Doubler. The key insight is the realization that the dueling bandits problem is an example of a *symmetric game* [Owen, 1995]. The Sparring algorithm uses separate MAB algorithms to control the choice of the different arms, which essentially reduces the conventional dueling bandits problem to two multi-armed bandit problems "sparring" against each other. This in turn is related to the *adversarial bandit* problem [Auer *et al.*, 2002], which has been the subject of extensive research [Bubeck and Cesa-Bianchi, 2012]. Although one can prove regret bounds using adversarial MAB algorithms (e.g., EXP3), empirically such algorithms perform much worse than using stochastic MAB algorithms.

Given an algorithm, $\mathcal{A}$, that solves the adversarial bandit problem, we can use it to solve the dueling bandits problem in the following fashion, called Sparring-$\mathcal{A}$: initiate a "row" copy of the algorithm, called $\mathcal{A}_r$, and a "column" copy, called $\mathcal{A}_c$; in each time-step, $\mathcal{A}_r$ proposes a "row" arm, which we denote by a_r, and $\mathcal{A}_c$ proposes a "column" arm, which we call a_c, and the two arms are compared against each other, with the probability of the row arm a_r beating the column arms a_c being p_{rc}; once the comparison has been carried out, the algorithm that proposed the arm that won the comparison receives a reward of 1 and the other side receives a reward of 0. In this setup, each copy of the algorithm plays the role of an adversary for the other.

The theory of adversarial bandits guarantees that if we make use of an adversarial bandit algorithm $\mathcal{A}$, then Sparring-$\mathcal{A}$ will incur regret of the form $\mathcal{O}(\sqrt{T})$, whereas the regret bounds proven for all of the algorithms discussed so far take the form $\mathcal{O}(\log T)$. What is intriguing, as far as the Sparring style of algorithms are concerned, is that extensive experimentation by various researchers has demonstrated that setting $\mathcal{A}$ to be a *non-adversarial* bandit algorithm like UCB, produces results that empirically attain a logarithmic regret rate [Ailon *et al.*, 2014].

Major improvements in this direction are presented in [Sui *et al.*, 2017b], where a related algorithm, called Self-Sparring, is analyzed with a $\mathcal{O}(K \log T)$ asymptotic regret bound. Self-Sparring uses a stochastic MAB algorithm such as Thompson sampling as a subroutine to independently sample the set of m arms, S_t to duel ($m = 2$ is the standard dueling bandits setting). Self-Sparring algorithm views the dueling bandits problem as a multi-player game with stochastic rewards and drifting dynamics. The high-level strategy is to reduce the multi-dueling bandits problem to a multi-armed bandit problem that can be solved using one single MAB

algorithm, and ideally lift existing MAB guarantees to the multi-dueling setting. Self-Sparring is a simple framework and easy to extend to continuous bandits by integrating with kernels to model dependencies across arms.

As numerous experimental investigations have demonstrated, Sparring & Self-Sparring tend to perform extremely well in practice. Therefore, the question of providing finite time $\mathcal{O}(K \log T)$ regret bounds for the symmetrically designed algorithms remain a very interesting open question.

4 Extensions and Applications

There are numerous extensions of the basic Condorcet dueling bandits setting that have been explored in the literature. We discuss these extensions in this section. The purpose of this effort has been to address the shortcomings of this setup, which include the following:

1. The Condorcet winner may fail to exist in practice. In such a scenario, there are numerous generalizations and substitutes for the Condorcet winner that could be employed.

2. In many applications, the preference matrix might not be fixed over time: it might depend on a context or might be set by an adversary.

3. In certain applications, it is feasible to carry out multiple comparisons simultaneously, so we would like to have algorithms that can make use of more complex feedback.

4. So far, we have treated each arm as a disjoint entity from the remaining arms, but oftentimes the set of arms comes equipped with certain structures that could be utilized to explore the arms more efficiently.

4.1 Beyond Condorcet Winners

What makes the Condorcet winner desirable as a solution concept is that one arm is unambiguously preferred to all other arms, which makes it easy to define a natural notion of regret. However, there is no guarantee that there exists an arm that is preferred to all other arms, as is the case in the rock-paper-scissors example. Indeed, the Condorcet winner can often not exist in practice [Zoghi *et al.*, 2015a], which requires new solution concepts for the dueling bandits problem.

The field of social choice theory has been grappling with situations where there is no Condorcet winner among the candidates and so it is unclear who the winner of, e.g., an election should be. Over the years, numerous definitions have been proposed to address this conundrum, and in more recent years corresponding dueling bandits algorithms have been proposed that converge to these solution concepts. We discuss these in the following together with a discussion of the advantages and disadvantages of each definition:

1. The *Borda winner* [Jamieson *et al.*, 2015] is the arm a_b with the largest *Borda score*, which is defined to be $\sum_j p_{bj}/K$: this is equivalent to the probability that a_b beats a uniformly randomly sampled arm a_j.
Even though the Borda winner always exists, it is not necessarily equal to the Condorcet winner if the latter exists, which is undesirable because the Condorcet winner is preferred to all other arms, including the Borda winner.

2. The *von Neumann winner* [Dudík *et al.*, 2015; Balsubramani *et al.*, 2016] is a probability distribution π_{vN} on the arms (rather than a single arm), which satisfies the property that if the arm a_i is sampled from π_{vN} and a_j is sampled from any other distribution on the arms, then $\mathbb{E}_{a_i,a_j}(p_{ij}) \geq 0.5$: in other words, arms sampled π_{vN} on average beat arms sampled from any other distribution.

The von Neumann winner is also guaranteed to exist by von Neumann's min-max theorem [Dudík *et al.*, 2015], rather than just the uniform distribution: indeed, if the Condorcet winner exists, then the von Neumann winner is the distribution that assigns all of its probability to the Condorcet winner. However, there might be certain situations where it is undesirable to adopt a solution concept that requires sampling from a distribution on the arms, e.g. due to memory constraints.

3. The *Copeland winner* [Zoghi *et al.*, 2015a; Komiyama *et al.*, 2016; Wu and Liu, 2016] is the arm with the highest *Copeland score*, which is the number of arms that a given arm beats on average. The Copeland winner is also guaranteed to exist and it coincides with the Condorcet winner if the latter exists, but unlike the von Neumann winner, the Copeland winner might lose to some other arms on average, which can be problematic if one would like to have a solution concept that has no weaknesses. Note that a Copeland winner is any arm with maximum Copeland score.

4. As with the set of Copeland winners, there exist other subsets of the arms that one could define based on the preference graph of the dueling bandits problem where we might consider arm in the set an acceptable solution. Examples of these include the Top Cycle, the Uncovered Set and the Banks Set, all of which collapse to the Condorcet winner if the latter exists: we refer the interested reader to [Ramamohan *et al.*, 2016] for the definitions of these sets.

[Chen and Frazier, 2017] studied the dueling bandit problem in the Condorcet winner setting, but with a different definition of regret. They consider a less well-studied form of regret, called weak regret, which is 0 if either arm pulled is the Condorcet winner. It proposes $WS - W$ algorithm that has expected cumulative weak regret that is $\mathcal{O}(N^2)$, and $\mathcal{O}(N \log(N))$ if arms have a total order.

4.2 Adversarial and Contextual Dueling Bandits

As also studied in the MAB problem setting, there are many settings where the parameters of the problem (i.e. the preference matrix) are not fixed over time. There are two problems settings in the dueling bandits literature that deal with such extensions: adversarial dueling bandits and contextual dueling bandits. In the case of the former, an adversary selects the preference matrix at each time-step. In the case of the contextual variant of the problem, the preference matrix at time t is determined by a context variable x_t that is drawn i.i.d. for all t from an unknown distribution, with only x_t being revealed to the algorithm; also, the algorithm has access to a pool of policies from which it needs to chose the optimal policy.

A simple solution to both of these problems is to use Sparring-EXP4, as discussed in Section 3, where EXP4 is an adversarial contextual bandit algorithm [Auer *et al.*, 2002], whose regret bound also provides a bound on the regret of Sparring-EXP4 in the adversarial contextual dueling bandits setting [Dudík *et al.*, 2015]. In the purely adversarial setting, Relative EXP3 (REX3) [Gajane *et al.*, 2015] has been pro-

posed as a modification to Sparring-EXP3, where the weights are shared between the two copies of EXP3.

In the contextual setting, one major drawback of EXP4 is that its computational complexity scales linearly with the number of available policies, which tends to be prohibitively large in practice. It is thus desirable to devise algorithms that make a limited number of calls to an oracle that produces the best policy given a set of past observations. This problem was partially addressed with the introduction of two algorithms, SparringFPL and ProjectedGD [Dudík *et al.*, 2015], which use an explore-first strategy, in the sense that given a certain time horizon, the algorithm allocated a certain number of time-steps to pure exploration and then it exploits using its best guess at the optimal policy afterwards. The drawback of this scheme is that the best attainable regret bound is of the form $\mathcal{O}(T^{2/3})$, whereas the regret bound of Sparring-EXP4 is $\mathcal{O}(\sqrt{T})$. Indeed, the problem of devising a computationally efficient contextual dueling bandits algorithm with optimal regret bound remains an interesting open problem.

4.3 With Multiple Comparisons

Recent attempts [Brost *et al.*, 2016; Sui *et al.*, 2017b] also extend the dueling bandits framework and proposed multi-dueling bandits algorithms for the intelligent selection of rankers for simultaneous comparisons and improves the trade-off between exploration and exploitation. [Brost *et al.*, 2016] provided the first multi-dueling algorithm. It is an empirical approach with upper confidence bound method as the subroutine. This work provided large scale experimental evaluations on both synthetic and real web search learning-to-rank datasets.

Algorithms that treat the arms asymmetrically are hard to extend to multi-dueling cases due to the asymmetric structure of picking arms. [Sui *et al.*, 2017b] addresses the multi-dueling bandits with symmetric sampling strategy for each arm. This makes the self-dueling Self-Sparring algorithm easy to be generated to multi-dueling. This work extends the original dueling bandits problem by simultaneously dueling multiple arms as well as modeling dependencies between arms using a kernel.

For this setting, the Self-Sparring algorithm algorithmically reduces the multi-dueling bandits problem into a conventional multi-armed bandit problem that can be solved using a stochastic bandit algorithm such as Thompson Sampling. It provides a regret analysis of the multi-dueling setting and guarantees the asymptotic regret to be $\mathcal{O}(K \ln T/\Delta)$. When multiple comparisons are feasible, multi-dueling algorithms yield orders of magnitude improvement in performance compared to conventional dueling bandits algorithms.

4.4 With Structured Input Spaces

Convex Dueling Bandits. [Yue and Joachims, 2009] proposes a dueling bandits gradient-descent method for optimizing information retrieval systems. It builds on methods for online convex optimization. The dueling bandits gradient descent approach is compatible with many existing classes of retrieval functions with theoretical guaranteed sublinear regret. [Kumagai, 2017] proposes a stochastic mirror descent algorithm and show that the algorithm achieves an $\mathcal{O}(\sqrt{T \log T})$ regret bound under strong convexity and smoothness assumptions for the cost function. It also shows the equivalence between regret minimization in dueling bandits and convex optimization for the cost function.

Sparse Dueling Bandits. [Jamieson *et al.*, 2015] proposes a new structural assumption for the K-armed dueling bandits problem in which the top arms can be distinguished by duels with a sparse set of other arms. An algorithm was developed for the dueling bandits problem under this assumption, with theoretical performance guarantees showing significant sample complexity improvements compared to naive reductions to standard multi-armed bandit algorithms.

Kernelized Dueling Bandits. [González *et al.*, 2017] proposes the approach aiming at combining the good properties of the dueling bandits methods with the advantages of having a probabilistic model able to capture correlations within the whole input space. Following the bandits settings, the key idea is to learn a preference function in the space of the duels by using a Gaussian process. This allows the agent to select the most relevant comparisons non-greedily. It is a pure Bayesian optimization approach without theoretical guarantees on convergence rate. [Sui *et al.*, 2017b] addresses two challenges in a unified framework, as multi-dueling bandits with dependent arms. This work extends the original dueling bandits problem by simultaneously dueling multiple arms as well as modeling dependencies between arms using a kernel. Explicitly formalizing these real-world characteristics provides an opportunity to develop principled algorithms that are much more efficient than algorithms designed for the original setting. Most dueling bandits algorithms suffer regret that scales linearly with the number of arms, which is not practical when the number of arms is very large or infinite. The Self-Sparring algorithm can incorporate dependencies using a Gaussian process prior with an appropriate kernel, and reduce the sample complexity from $\mathcal{O}(K)$ to $\mathcal{O}(d)$ where d is the dimension of the kernel. [Sui *et al.*, 2018] extended this idea by incorporating dueling bandits within safe Bayesian optimization. It solves the optimization of an unknown utility function with absolute feedback or preference feedback subject to unknown safety constraints.

4.5 Applications

Due to the ubiquity of preference elicitation, the dueling bandit setting enjoys a broad range of applications, in both preference-based optimization in other theoretical settings and real-world applications.

Preference-based reinforcement learning is considered in [Fürnkranz *et al.*, 2012]. This approach is basically a preference-based racing algorithm that selects the best among a given set of candidate policies with high probability. The algorithm operates on a suitable ordinal preference structure and only uses pairwise comparisons between sample roll-outs of the policies. This work provides both formal performance and complexity analysis and experimental studies for the effectiveness and efficiency of the algorithm. Some related

works on preference-based reinforcement learning were surveyed in [Wirth *et al.*, 2017].

The problem of online rank elicitation with dueling bandits is studied in [Szörényi *et al.*, 2015]. It assumes that the rankings of a set of alternatives obey the Plackett-Luce distribution, a widely used probability distribution over rankings. Following the setting of the dueling bandits problem, the learner is allowed to query pairwise comparisons between alternatives. This work provides a formal complexity analysis of the algorithms and experimental studies showing the effectiveness in practice. A variant of structured dueling bandits setting also shows the improvement over machine translation tasks [Sokolov *et al.*, 2016].

[Sui and Burdick, 2014] proposed a Rank-Comparison algorithm to efficiently solve a specific bandit problem using subgroup rank feedback. The application of this algorithm is efficiently optimize therapies within a restricted action space for clinical treatment. [Sui *et al.*, 2017a] presents the first time an online learning algorithm was applied towards spinal cord injury treatments. The paraplegic human patients could achieve full-weight standing under the stimulation strategies provided by the algorithm. The effectiveness and efficiency of dueling bandits approach is recognized in clinical treatments as shown in the paper. [Sui *et al.*, 2018] presents the safe optimization of clinical neurological therapies.

In the area of search engine optimization, it has been shown that pairwise comparisons between pairs of documents can lead to great improvements in the quality of the search results [Zoghi *et al.*, 2016], so an interesting question is if online ranking methods [Zoghi *et al.*, 2017] can be adapted to use dueling bandits, rather than MAB methods.

Another way that dueling bandits have been used to improved search engines has been through the use of interleaved comparisons [Radlinski *et al.*, 2008; Hofmann *et al.*, 2011; Radlinski and Craswell, 2013] which is a method for comparing two rankers (e.g. Google vs Bing). The comparison is carried out by submitting a query issued by a user to both systems to get two results lists and then interleaving the two ranked lists before presenting the merged list to the user. Once the user's interactions have been received, then we can infer which result list was preferred by the user. This method can be used both for the sake of evaluating multiple proposed improvements to a ranker [Schuth *et al.*, 2015] and for learning the ranker itself [Hofmann *et al.*, 2013].

5 Conclusions and Outlook

This paper provides a survey of the theories, algorithms, and applications of the dueling bandits problem. Dueling bandits is an emerging field of research. In contrast to standard MAB problems, where feedback comes in the form of (stochastic) real valued rewards produced by the arms, dueling bandits setting has indirect preference feedback. We have given an overview of the dueling bandits problem that have been studied in the literature, algorithms for tackling them, and applications that could be benefit from such algorithms.

While this research area is still in its early stages, some of the recent advancements have already been successfully applied in concrete applications, such as information retrieval, search engine improvement, and clinical online recommendation. As reinforcement learning can be viewed as a stateful generalization of the bandit problem, the dueling bandits problem can also be a key component for preference based reinforcement learning. With this survey, we hope to have presented a clear background and encourage further fundamental and applied research in this area.

References

[Ailon *et al.*, 2014] Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *ICML*, 2014.

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 2002.

[Balsubramani *et al.*, 2016] Akshay Balsubramani, Zohar Karnin, Robert E. Schapire, and Masrour Zoghi. Instance-dependent regret bounds for dueling bandits. In *COLT*, 2016.

[Brost *et al.*, 2016] Brian Brost, Yevgeny Seldin, Ingemar J Cox, and Christina Lioma. Multi-dueling bandits and their application to online ranker evaluation. In *CIKM*, 2016.

[Bubeck and Cesa-Bianchi, 2012] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012.

[Busa-Fekete and Hüllermeier, 2014] Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *ALT*. Springer, 2014.

[Cesa-Bianchi and Lugosi, 2006] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[Cesa-Bianchi *et al.*, 2006] Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Regret minimization under partial monitoring. *Mathematics of Operations Research*, 31(3):562–580, 2006.

[Chapelle *et al.*, 2012] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems*, 2012.

[Chen and Frazier, 2017] Bangrui Chen and Peter I. Frazier. Dueling bandits with weak regret. In *ICML*, 2017.

[Dudík *et al.*, 2015] Miroslav Dudík, Katja Hofmann, Robert E Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *COLT*, 2015.

[Fürnkranz *et al.*, 2012] Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning*, 89(1-2):123–156, 2012.

[Gajane *et al.*, 2015] Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *ICML*, 2015.

[González *et al.*, 2017] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential bayesian optimization. In *ICML*, 2017.

[Hofmann *et al.*, 2011] Katja Hofmann, Shimon Whiteson, and Maarten De Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM*, 2011.

[Hofmann *et al.*, 2013] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. Reusing historical interaction data for faster online learning to rank for ir. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 183–192. ACM, 2013.

[Jamieson *et al.*, 2015] Kevin Jamieson, Sumeet Katariya, Atul Deshpande, and Robert Nowak. Sparse dueling bandits. In *AISTATS*, 2015.

[Komiyama *et al.*, 2015] Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In *COLT*, 2015.

[Komiyama *et al.*, 2016] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Copeland dueling bandit problem: Regret lower bound, optimal algorithm, and computationally efficient algorithm. In *ICML*, 2016.

[Kumagai, 2017] Wataru Kumagai. Regret analysis for continuous dueling bandit. In *NIPS*, 2017.

[Owen, 1995] Guillermo Owen. *Game Theory*. Emerald Group Publishing Limited, 3rd edition, 1995.

[Radlinski and Craswell, 2013] Filip Radlinski and Nick Craswell. Optimized interleaving for online retrieval evaluation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 245–254. ACM, 2013.

[Radlinski *et al.*, 2008] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008.

[Ramamohan *et al.*, 2016] Siddartha Y Ramamohan, Arun Rajkumar, and Shivani Agarwal. Dueling bandits: Beyond condorcet winners to general tournament solutions. In *NIPS*, 2016.

[Robbins, 1952] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 1952.

[Schuth *et al.*, 2015] Anne Schuth, Katja Hofmann, and Filip Radlinski. Predicting search satisfaction metrics with interleaved comparisons. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 463–472. ACM, 2015.

[Sokolov *et al.*, 2016] Artem Sokolov, Stefan Riezler, and Tanguy Urvoy. Bandit structured prediction for learning from partial feedback in statistical machine translation. In *MT summit*, 2016.

[Sui and Burdick, 2014] Yanan Sui and Joel Burdick. Clinical online recommendation with subgroup rank feedback. In *RecSys*. ACM, 2014.

[Sui *et al.*, 2017a] Yanan Sui, Yisong Yue, and Joel Burdick. Correlational dueling bandits with application to clinical treatment in large decision spaces. In *IJCAI*, 2017.

[Sui *et al.*, 2017b] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *UAI*, 2017.

[Sui *et al.*, 2018] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *ICML*, 2018.

[Szörényi *et al.*, 2015] Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for plackett-luce: A dueling bandits approach. In *NIPS*, 2015.

[Urvoy *et al.*, 2013] Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In *ICML*, 2013.

[Wirth *et al.*, 2017] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

[Wu and Liu, 2016] Huasen Wu and Xin Liu. Double thompson sampling for dueling bandits. In *NIPS*, 2016.

[Yue and Joachims, 2009] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML*, 2009.

[Yue and Joachims, 2011] Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *ICML*, 2011.

[Yue *et al.*, 2012] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 2012.

[Zoghi *et al.*, 2014a] Masrour Zoghi, Shimon Whiteson, Maarten de Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *WSDM*, 2014.

[Zoghi *et al.*, 2014b] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *ICML*, 2014.

[Zoghi *et al.*, 2015a] Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. In *NIPS*, 2015.

[Zoghi *et al.*, 2015b] Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *WSDM*, 2015.

[Zoghi *et al.*, 2016] Masrour Zoghi, Tomáš Tunys, Lihong Li, Damien Jose, Junyan Chen, Chun Ming Chin, and Maarten de Rijke. Click-based hot fixes for underperforming torso queries. In *SIGIR*, 2016.

[Zoghi *et al.*, 2017] Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *ICML*, 2017.