Journal of Machine Learning Research 22 (2021) 1-108 Submitted 8/18; Revised 12/19; Published 01/21

Preference-based Online Learning with Dueling Bandits: A Survey

Viktor Bengs VIKTOR.BENGS@UPB.DE
Heinz Nixdorf Institute and Department of Computer Science
Paderborn University, Germany

Róbert Busa-Fekete BUSAROBI@GOOGLE.COM
Google Research
New York, NY, USA

Adil El Mesaoudi-Paul ADIL.PAUL@UPB.DE
Heinz Nixdorf Institute and Department of Computer Science
Paderborn University, Germany

Eyke Hüllermeier EYKE@UPB.DE
Heinz Nixdorf Institute and Department of Computer Science
Paderborn University, Germany

Editor: Peter Auer

Abstract

In machine learning, the notion of *multi-armed bandits* refers to a class of online learning problems, in which an agent is supposed to simultaneously explore and exploit a given set of choice alternatives in the course of a sequential decision process. In the standard setting, the agent learns from stochastic feedback in the form of real-valued rewards. In many applications, however, numerical reward signals are not readily available—instead, only weaker information is provided, in particular relative preferences in the form of qualitative comparisons between pairs of alternatives. This observation has motivated the study of variants of the multi-armed bandit problem, in which more general representations are used both for the type of feedback to learn from and the target of prediction. The aim of this paper is to provide a survey of the state of the art in this field, referred to as preference-based multi-armed bandits or dueling bandits. To this end, we provide an overview of problems that have been considered in the literature as well as methods for tackling them. Our taxonomy is mainly based on the assumptions made by these methods about the data-generating process and, related to this, the properties of the preference-based feedback.

Keywords: Multi-armed bandits, online learning, preference learning, ranking, top-k selection, exploration/exploitation, cumulative regret, sample complexity, PAC learning

1. Introduction

Multi-armed bandit (MAB) algorithms have received considerable attention and have been studied quite intensely in machine learning in the recent past. The great interest in this topic is hardly surprising, given that the MAB setting is not only theoretically challenging but also practically useful, as can be seen from its use in a wide range of applications. For example, MAB algorithms turned out to offer effective solutions for problems in medical treatment design (Lai and Robbins, 1985; Kuleshov and Precup, 2014), online advertisement

(Chakrabarti et al., 2008), and recommendation systems (Kohli et al., 2013), just to mention a few.

The multi-armed bandit problem, or bandit problem for short, is one of the simplest instances of the sequential decision making problem, in which a *learner* (also called decision maker or agent) needs to select *options* from a given set of alternatives repeatedly in an online manner—referring to the metaphor of the eponymous gambling machine in casinos, these options are also associated with "arms" that can be "pulled". More specifically, the agent selects one option at a time and observes a numerical (and typically noisy) *reward* signal providing information on the quality of that option. The goal of the learner is to optimize an evaluation criterion such as the *error rate,* i.e., the expected percentage of suboptimal pulls, or the *cumulative regret*, i.e., the expected difference between the sum of rewards that could have been obtained by playing the best arm (defined as the one generating the highest rewards on average) in each round and the sum of the rewards actually obtained. To achieve the desired goal, the online learner has to cope with the famous exploration/exploitation dilemma (Auer et al., 2002a; Cesa-Bianchi and Lugosi, 2006; Lai and Robbins, 1985): It has to find a reasonable compromise between playing the arms that produced high rewards in the past (exploitation) and trying other, possibly even better arms the (expected) rewards of which are not precisely known so far (exploration).

The assumption of a numerical reward signal is a potential limitation of the MAB setting. In fact, there are many practical applications in which it is hard or even impossible to quantify the quality of an option on a numerical scale. More generally, the lack of precise feedback or exact supervision has been observed in other branches of machine learning, too, and has led to the emergence of fields such as *weakly supervised learning* and *preference learning* (Fürnkranz and Hüllermeier, 2011). In the latter, feedback is typically represented in a purely qualitative way, namely in terms of pairwise comparisons or rankings. Feedback of this kind can be useful in online learning, too, as has been shown in online information retrieval (Hofmann, 2013; Radlinski et al., 2008). As another example, think of crowd-sourcing services like the Amazon Mechanical Turk, where simple questions such as pairwise comparisons between decision alternatives are asked to a group of annotators. The task is to approximate an underlying target ranking on the basis of these pairwise comparisons, which are possibly noisy and partially non-coherent (Chen et al., 2013). Another application worth mentioning is the ranking of XBox gamers based on their pairwise online duels; the ranking system of XBox is called TrueSkill$^{\text{TM}}$ (Guo et al., 2012).

Extending the multi-armed bandit setting to the case of preference-based feedback, i.e., the case in which the online learner is allowed to compare arms in a qualitative way, is therefore a promising idea. And indeed, extensions of that kind have received increasing attention in the recent years. The aim of this paper is to provide a survey of the state of the art in the field of preference-based multi-armed bandits (PB-MAB); it updates and significantly extends an earlier review by Busa-Fekete and Hüllermeier (2014). After recalling the basic setting of the problem in Section 2, we provide an overview of methods that have been proposed to tackle PB-MAB problems in Sections 3 and 4. Our taxonomy is mainly based on the assumptions made by these methods about the data-generating process or, more specifically, the properties of the pairwise comparisons between arms. Our survey is focused on the *stochastic* MAB setup, in which feedback is generated according to an underlying (unknown but stationary) probabilistic process; we do not cover the case of *adversarial* data-generating processes, which has recently received a lot of attention as well (Ailon et al., 2014a; Dudík et al., 2015; Gajane and Urvoy, 2015; Zimmert and Seldin, 2019), except briefly in Section 5, where also other extensions of the basic PB-MAB problem are discussed. Due to the surge of research interest in the multi-dueling variant of the basic PB-MAB problem in the recent past[1], we devote Section 6 to this extension. Finally, we highlight some interesting practical applications of the PB-MAB problem in Section 7 prior to concluding with a discussion of open issues within the scope of the survey in Section 8.

2. The Basic Preference-based Multi-Armed Bandit Problem

The stochastic MAB problem with pairwise comparisons as actions has been studied under the notion of "dueling bandits" in several papers (Yue and Joachims, 2009; Yue et al., 2012). Although this term has been introduced for a concrete setting with specific modeling assumptions (Sui et al., 2018b), it is meanwhile used more broadly for variants of that setting, too. Throughout this paper, we shall use the terms "dueling bandits" and "preference-based bandits" synonymously, although the latter may indeed seem more convenient in light of the recent multi-dueling variant (cf. Section 6).

Consider a fixed set of arms (options) $\mathcal{A} = \{a_1, \ldots, a_K\}$. As actions, the learning algorithm (or simply the learner or agent) can perform a comparison between any pair of arms a_i and a_j, i.e., the action space can be identified with the set of index pairs (i, j) such that $1 \leq i \leq j \leq K$. We assume the feedback observable by the learner to be generated by an underlying (unknown) probabilistic process characterized by a *preference relation*

$$\mathbf{Q} = [q_{i,j}]_{1 \leq i,j \leq K} \in [0,1]^{K \times K} \ .$$

More specifically, for each pair of actions (a_i, a_j), this relation specifies the pairwise preference probability

$$\mathbf{P}\left(a_i \succ a_j\right) = q_{i,j} \tag{1}$$

of observing a preference for a_i in a direct comparison with a_j. Thus, each $q_{i,j}$ specifies a Bernoulli distribution. These distributions are assumed to be stationary and independent, both across actions and iterations. Thus, whenever the learner takes action (i, j), the outcome is distributed according to (1), regardless of the outcomes in previous iterations.

The relation $\mathbf{Q}$ is reciprocal in the sense that $q_{i,j} = 1 - q_{j,i}$ for all $i, j \in [K] := \{1, \ldots, K\}$. We note that, instead of only observing strict preferences, one may also allow a comparison to result in a *tie* or an *indifference*. In that case, the outcome is a trinomial instead of a binomial event. Since this generalization makes the problem technically more complicated, though without changing it conceptually, we shall not consider it further. Busa-Fekete et al. (2013, 2014b) handle indifference by giving "half a point" to both arms, which, in expectation, is equivalent to deciding the winner by tossing a coin. Thus, the problem is essentially reduced to the case of binomial outcomes.

We say arm a_i beats arm a_j if $q_{i,j} > 1/2$, i.e., if the probability of winning in a pairwise comparison is larger for a_i than it is for a_j. Clearly, the closer $q_{i,j}$ is to $1/2$, the harder it

1. This serves as an additional motivation for referring to both settings, i.e, dueling bandits and multi-dueling bandits, as the *preference-based multi-armed bandits* such as implicitly proposed by this survey, since this notion unifies the latter two closely related fields in a quite intuitive way.

becomes to distinguish the arms a_i and a_j based on a finite sample set from $\mathbf{P}(a_i \succ a_j)$. In the worst case, when $q_{i,j} = 1/2$, one cannot decide which arm is better based on a finite number of pairwise comparisons. Therefore,

$$\Delta_{i,j} = q_{i,j} - \frac{1}{2}$$

appears to be a reasonable quantity to characterize the hardness of a PB-MAB task (whatever goal the learner wants to achieve), which we call the *calibrated pairwise preference probabilities*. Note that $\Delta_{i,j}$ can also be negative (opposed to the value-based setting, in which the usual quantity used for characterizing the complexity of a multi-armed bandit task is always positive and depends on the gap between the means of the best arm and the suboptimal arms).

2.1 Learning Protocol

The decision making process iterates in discrete steps, either through a finite time horizon $\mathbb{T} := [T] = \{1, \ldots, T\}$ or an infinite horizon $\mathbb{T} := \mathbb{N}$. As mentioned above, the learner is allowed to compare two actions in each iteration $t \in \mathbb{T}$. Thus, in each iteration t, it selects an index pair $1 \leq i(t) \leq j(t) \leq K$ and observes

$$\begin{cases} a_{i(t)} \succ a_{j(t)}, & \text{with probability } q_{i(t),j(t)} \\ a_{j(t)} \succ a_{i(t)}, & \text{with probability } q_{j(t),i(t)} \end{cases} .$$

The pairwise probabilities $q_{i,j}$ can be estimated on the basis of finite sample sets. Consider the set of time steps among the first t iterations, in which the learner decides to compare arms a_i and a_j, and denote the size of this set by $n^t_{i,j}$. Moreover, denoting by $w^t_{i,j}$ and $w^t_{j,i}$ the frequency of "wins" of a_i and a_j, respectively, the proportion of wins of a_i against a_j up to iteration t is then given by

$$\widehat{q}^{\,t}_{i,j} = \frac{w^t_{i,j}}{n^t_{i,j}} = \frac{w^t_{i,j}}{w^t_{i,j} + w^t_{j,i}} .$$

Since our samples are assumed to be independent and identically distributed (i.i.d.), $\widehat{q}^{\,t}_{i,j}$ is a plausible estimate of the pairwise winning probability defined in (1). Yet, this estimate might be biased, since $n^t_{i,j}$ depends on the choice of the learner, which in turn depends on the data; therefore, $n^t_{i,j}$ itself is a random quantity. A high probability confidence interval $c^t_{i,j}$ for $q_{i,j}$ can be obtained based on the Hoeffding's bound (Hoeffding, 1963), which is commonly used in the bandit literature. Although the specific computation of the confidence intervals may differ from case to case, they are generally of the form $[\widehat{q}^{\,t}_{i,j} \pm c^t_{i,j}]$. Accordingly, if $\widehat{q}^{\,t}_{i,j} - c^t_{i,j} > 1/2$, arm a_i beats arm a_j with high probability; analogously, a_i is beaten by arm a_j with high probability, if $\widehat{q}^{\,t}_{i,j} + c^t_{i,j} < 1/2$.

2.2 Learning Tasks

The notion of optimality of an arm is far less obvious in the preference-based setting than it is in the value-based (numerical) setting. In the latter, the optimal arm is (usually) simply the one with the highest expected reward—more generally, the expected reward induces a natural total order on the set of arms $\mathcal{A}$. In the preference-based case, the connection between the pairwise preferences $\mathbf{Q}$ and the order induced by this relation on $\mathcal{A}$ is less trivial; in particular, the latter may contain preferential cycles. In the following, we provide an overview of different notions of (sub-)optimality of arms and the related learning tasks.

2.2.1 Best Arm

In the preference-based setting, it appears natural to define the best arm as the one that is preferred to any other arm. Formally, $a_{i^*} \in \mathcal{A}$ is the best arm if $\Delta_{i^*,j} > 0$ for all $j \in [K]\backslash\{i^*\}$. This definition corresponds to the definition of the so-called *Condorcet winner.* In spite of being natural, it comes with the major drawback that a Condorcet winner is not guaranteed to exist for every preference relation $\mathbf{Q}$. Due to this problem, various alternative notions for a best arm have been suggested, each with its own advantages and drawbacks. These different alternatives will be introduced in Section 4. For sake of simplicity, we subsequently assume the existence of a Condorcet winner and denote it by a_{i^*} throughout the rest of this section.

2.2.2 Rankings of Arms

Another target for the learner, more ambitious than merely finding an optimal arm, is to find an entire ranking over the arms. A ranking can be identified by a permutation $\pi : \{1, \ldots, K\} \to \{1, \ldots, K\}$, with $\pi(i)$ specifying the rank of arm $a_i \in \mathcal{A}$ (and $\pi^{-1}(i)$ the index of the arm on the i^{th} position). For a sensible definition of a target ranking of the arms, similar issues arise like in the specification of the best arm.

Indeed, given a preference relation $\mathbf{Q}$, the arguably most natural approach is to consider the ranking of the arms such that arm a_i is ranked higher than another arm a_j if and only if the former beats the latter ($\Delta_{i,j} > 0$). However, just like the Condorcet winner may fail to exist for a given preference relation $\mathbf{Q}$, such a ranking may not exist due to preferential cycles. Again, to circumvent this problem, alternative definitions of target rankings have been proposed in the literature (cf. Section 4).

2.2.3 Top-positioned Arms

There are several applications in which the entire ranking over the arms is not of major relevance. Instead, only the ordering or merely the identity of the top-k arms, i.e., the best k elements of the underlying full ranking (assuming this ranking to exist), are of interest. This leads to the problems of *top-k ranking* and *top-k identification*

In general, top-k identification is an easier learning task, as it requires less information than the top-k ranking (the former can be derived from the latter, but not the other way around). There are, however, examples of learning tasks for which both targets have the same complexity.

2.2.4 Estimation of the Preference Relation

The problem of estimating the preference relation $\mathbf{Q}$, which has been tackled as an offline learning task under the assumption of certain structural properties on $\mathbf{Q}$ (Shah et al., 2016), can also be considered in an online learning framework. A related target emerges by assuming that the preference probabilities in (1) are the marginals of a distribution over rankings (cf. Section 3.3), and one seeks to estimate the entire ranking distribution based on these marginals.

2.2.5 Near-optimal Targets

There are various applications in which it suffices to produce a reasonably good approximation of the truly optimal target. Yet, compared to the classical MAB setting, approximation errors are less straightforward to define in the preference-based setup, again due to the lack of numerical rewards. In spite of this, for many of the just introduced targets, there have been attempts to define reasonable surrogates (Falahatgar et al., 2017a,b, 2018).

For $\epsilon \in (0, 1/2)$, an arm a_i is called *ϵ-preferable* to an arm a_j if $\Delta_{i,j} \geq -\epsilon$. Equipped with this notion, one can define an *ϵ-optimal arm* as an arm that is *ϵ-preferable* to the optimal arm a_{i^*}.

If the target is the natural ranking introduced in Section 2.2.2, which we shall identify by a permutation π^* on $[K]$ such that $\pi^*(i) > \pi^*(j)$ implies $\Delta_{i,j} > 0$, then an *ϵ-optimal* variant of π^* is any permutation π on $[K]$ such that $\pi(i) > \pi(j)$ implies that a_i is ϵ-preferable to arm a_j. For target rankings other than π^*, it is possible to define *ϵ-optimal* variants in a similar way, for example by specifying a sensible metric d on the space of all permutations and calling ϵ-optimal each ranking that is ϵ-close to the target permutation with respect to d. A more detailed discussion is deferred to Section 4.

2.3 Performance Measures

As usual in the realm of machine learning, there are different goals a learner may pursue, and correspondingly different ways to quantify the overall performance of a learner. The most prominent goals and performance measures will be discussed in the following.

2.3.1 Regret Bounds

In a preference-based setting, defining a reasonable notion of regret is not as straightforward as in the value-based setting, where the sub-optimality of an action can be expressed easily on a numerical scale. In particular, since the learner selects two arms to be compared in an iteration, the sub-optimality of both of these arms should be taken into account. A commonly used definition of regret is the following (Yue and Joachims, 2009, 2011; Urvoy et al., 2013; Zoghi et al., 2014b): Suppose the learner selects arms $a_{i(t)}$ and $a_{j(t)}$ in time step t, then its regret per time is

$$r_{t,avg} := \frac{\Delta_{i^*,i(t)} + \Delta_{i^*,j(t)}}{2} \quad ,$$

i.e., the average quality (with respect to the target arm a_{i^*}) of the chosen arms in t, to which we refer as the *average regret.* The *cumulative regret* incurred by the learner A up to time T is then given by

$$R_A^T := \sum_{t=1}^{T} r_{t,avg} = \sum_{t=1}^{T} \frac{\Delta_{i^*,i(t)} + \Delta_{i^*,j(t)}}{2} \quad . \tag{2}$$

For sake of brevity, we will suppress the dependency on A in the notation of the cumulative regret and simply write R^T if the learner is clear from the context. This regret takes into account the optimality of both arms, meaning that the learner has to select two nearly optimal arms to incur small regret. Note that this regret is zero if the optimal arm a_{i^*} is compared to itself, i.e., if the learner effectively abstains from gathering further information and instead fully commits to the arm a_{i^*}.

The regret defined in (2) reflects the average quality of the decision made by the learner. Obviously, one can define a more strict resp. less strict regret by taking the maximum or minimum of the calibrated pairwise probabilities, respectively, instead of their average. Formally, the *strong and weak regret* in time step t are defined, respectively, as

$$r_{t,\max} := \max\left\{\Delta_{i^*,i(t)}, \Delta_{i^*,j(t)}\right\} \quad , \quad r_{t,\min} := \min\left\{\Delta_{i^*,i(t)}, \Delta_{i^*,j(t)}\right\} \quad .$$

Obviously, it holds that $r_{t,\min} \le r_{t,avg} \le r_{t,\max}$, so that one has a hierarchy among these notions of regret. Furthermore, note that some works refer to the average regret $r_{t,avg}$ also as the strong regret, which is due to the (arti-)fact that the average regret is zero only if both $a_{i(t)}$ and $a_{j(t)}$ correspond to the target arm a_{i^*}. This peculiarity of the average regret is a motivation for allowing $a_{i(t)} = a_{j(t)}$, which can be interpreted as a *full commitment* to a single arm usually adopted as a "final" action, once being sure enough that the target arm has been found.

From a theoretical point of view, a distinction between these regret definitions is important, as shown by Chen and Frazier (2017) and further discussed in Section 3.1.8. In the rest of the paper, when speaking about the regret of a learner, we will refer to the regret in (2) with respect to $r_{t,avg}$ unless otherwise stated.

Another notion of regret per time is considered in the literature if the pairwise probabilities are modeled by utility functions, which will be discussed in Section 3.2. However, this additional notion can be expressed as a linear transformation of the average regret $r_{t,avg}$, and consequently only scales the cumulative regret.

In a theoretical analysis of an MAB algorithm, one is typically interested in providing a bound on the (cumulative) regret produced by that algorithm. We are going to distinguish two types of regret bound. The first one is the *expected regret bound*, which is of the form

$$\mathbf{E}\left[R^T\right] \le B(\mathbf{Q}, K, T) \ , \tag{3}$$

where $\mathbf{E}\left[\cdot\right]$ is the expected value operator, R^T is the regret accumulated till time step T, and $B(\cdot)$ is a positive real-valued function with the following arguments: the pairwise probabilities $\mathbf{Q}$, the number of arms K, and the iteration number T. This function may additionally depend on parameters of the learner, for example a tuning- or hyper-parameter. However, we neglect this dependence here. The expectation is taken with respect to the stochastic nature of the data-generating process and the (possible) internal randomization of the online learner. The regret bound (3) is technically akin to the expected regret bound of value-based multi-armed bandit algorithms like the one that is calculated for UCB (Auer et al., 2002a), although the parameters used for characterizing the complexity of the learning task are different.

The bound in (3) does not inform about how the regret achieved by the learner is concentrated around its expectation. Therefore, one might prefer to consider a second type of a reasonable regret bound, namely one that holds with high probability. This bound can be written in the form

$$\mathbf{P}\Big(R^T < B(\mathbf{Q}, K, T, \delta) \Big) \geq 1 - \delta \ .$$

For simplicity, we also say that the regret achieved by the online learner is $\mathcal{O}(B(\mathbf{Q}, K, T, \delta))$ with high probability.

Similar to the classical problem of online learning, if the goal is to minimize the regret, one is typically interested in *no-regret* algorithms, i.e., algorithms the regret bound function B of which grows sublinearly in the time horizon T, if the remaining components are fixed. In general, there are two types of regret bounds. First, the *gap-dependent regret bounds* depending on the calibrated preference probabilities with respect to the best arm. These bounds typically grow logarithmically with the time horizon T. Second, the *gap-independent regret bounds*, which are typically given as a specific root function of the time horizon T, but opposed to the first variant do not depend on the calibrated preference probabilities, though still on the number of arms K. The latter type of regret bounds corresponds in general to the worst-case learning scenario.

2.3.2 PAC Setting

In many applications, one is willing to gain efficiency at the cost of optimality: The algorithm is allowed to return a solution that is only approximately optimal, though it is supposed to do so more quickly. The variable of interest is then the *sample complexity of the learner,* that is the number of pairwise comparisons it queries prior to termination for returning a nearly optimal target (cf. Section 2.2). Such settings are referred to as *probably approximately correct* (PAC) settings (Even-Dar et al., 2002) and have been studied extensively for the classical MAB problem (Mannor and Tsitsiklis, 2004; Bubeck and Cesa-Bianchi, 2012).

A preference-based MAB algorithm is called (ϵ, δ)-PAC preference-based MAB algorithm with a *sample complexity* $B(\mathbf{Q}, K, \epsilon, \delta)$, if it terminates and returns an ϵ-optimal target[2] with probability at least $1 - \delta$, and the number of comparisons taken by the algorithm is at most $B(\mathbf{Q}, K, \epsilon, \delta)$. Note that only the number of the pairwise comparisons is taken into account, which means that pairwise comparisons are equally penalized, independently of the suboptimality of the arms chosen; in this regard, the setting differs from the goal of regret minimization.

2.3.3 Exact Sample Complexity

The exact sample complexity analysis is the strict version of the PAC learning goal described above, where the learner is supposed to return the correct target instead of a nearly optimal target. This setting is sometimes also called the δ-PAC setting (Kaufmann et al., 2016). Obviously, when the allowed approximation error ϵ of the (ϵ, δ)-PAC learning scenario tends to zero, these two settings coincide. However, while in the PAC learning scenario the bound on the sample complexity focuses on the approximation error ϵ coming from the extrinsic problem formulation, the bounds for the exact sample complexity analysis focuses

2. Definitions of ϵ-optimal targets are less straightforward in the PB-MAB problem. Such definitions will be given in subsequent sections.

on the intrinsic hardness of the problem represented by the smallest calibrated preference probability. Hence, an upper bound on the exact sample complexity of a learner also provides an upper bound on its sample complexity in the PAC setting.

2.4 Algorithm Classes

Despite the recency of the field of PB-MAB problems, a striking variety of algorithms has been developed to tackle the different targets and goals described above. Many of these algorithms are based on similar ideas and essentially invoke the same learning principles, such as efficient sorting or the derivation of representative statistics. In the following, we propose a categorization of existing algorithms into different algorithm classes, each of them characterized by specific properties. Note, however, that the algorithm classes are not mutually exclusive, since some learning algorithms are combining essential concepts from different classes at the same time.

2.4.1 MAB-related Algorithms

As the PB-MAB problem is a variant of the classical MAB problem, it seems quite natural to exploit established algorithmic ideas for the latter in order to construct algorithms for the former—hoping, of course, to preserve corresponding benefits. This connection is established in the existing methods in two ways:

– *Reduction to MAB problems:* The dueling bandits problem can be interpreted as a symmetric zero-sum game between two players (Owen, 1982), in the sense that one player always pulls the "left" arm of the duel and the other always the "right" arm. The winner of the duel gains a reward of one and the other a loss of one. Thus, using two classical MAB algorithms to determine the choice of a player, respectively, results in a conversion of the dueling bandits problem into some sort of a meta-MAB problem allowing the transfer of well-established theoretical guarantees.

– *Generalization of MAB algorithms:* At the heart of the PB-MAB problem is the estimation of the pairwise preference probabilities $q_{i,j}$, as these encode the quality of the arms in a similar manner as the rewards in the MAB problem. Thus, another natural approach, especially for the task of regret minimization, is to adapt the different high-level ideas of MAB algorithms revolving around an appropriate estimation of the rewards in order to strike a balance between exploration and exploitation for the pairwise preference probabilities. For this purpose, several well-established concepts are available: the optimism in face of uncertainty principle most prominently represented by UCB-type algorithms (Auer et al., 2002b), the probability matching heuristic underlying Thompson sampling (Thompson, 1933), the value-based racing task (Maron and Moore, 1994, 1997), and the explore-then-exploit principle (Robbins, 1952).

2.4.2 Online Optimization-based Algorithms

Under specific assumptions on the data-generating process as determined by the preference relation $\mathbf{Q}$ as well as on the set of arms, it is possible to cast the problem as a classical online optimization problem. For problems of that kind, powerful online learning algorithms are available (Shalev-Shwartz, 2012; Hazan, 2016), which typically exploit properties such as convexity of the target (function).

2.4.3 Noisy-sorting Algorithms

If the target of the learner is to provide a ranking over the arms, the most natural approach would be to sort the arms according to their optimality[3], giving rise to the class of noisy-sorting algorithms (Ailon et al., 2005). The active sampling strategies underlying such algorithms mimic the behavior of efficient sorting algorithms, such as merge sort or quicksort. The main difference between deterministic sorting and these sampling strategies is that, due to the assumed stochasticity of the observed feedback, the order between two arms can only be determined with a certain probability and requires repeated comparisons. Moreover, to guarantee representativeness of the observed comparisons for the target ranking, one has to require certain regularity assumptions, as will be thoroughly discussed in the subsequent sections.

2.4.4 Tournament Algorithms

Another approach for the task of finding an optimal arm or to identify the top-k arms is based on the concept of tournament systems as commonly considered in sports and gaming. Just like in the various sports disciplines, there are different types of tournament styles a tournament algorithm can employ. Yet, from a high-level point of view, all these algorithms are proceeding as follows. First, all arms are divided into groups, and duels are successively conducted among the arms within one group. This phase is usually called a "round". At the end of each round, the size of the group is diminished by discarding the inferior arms, and some of the groups are merged. The procedure of rounds is repeated until the size of the target is reached through successive elimination. The main differences for algorithms of this type lie in the decomposition of the arms into groups (the group sizes), the order in which duels within a group are played, the duration of a round, and the merging procedure for the groups after each round.

2.4.5 Challenge Algorithms

Just like tournament algorithms are inspired by tournament systems in sports, challenge algorithms are inspired by the challenge system most prominently represented by the World Chess Championship. Transferring the idea to the dueling bandits problem, an algorithm of the challenge-type keeps a reference arm (the current champion) as well as a set of comparison arms (the challengers), and compares the reference arm systematically with different comparison arms. This is done until either a challenger is defeated by the reference arm and discarded from the set of comparison arms, or the reference arm is defeated by a comparison arm, whereupon the latter becomes the new reference arm. In contrast to the challenge system in sports, it is allowed that the challenger arm might vary in each round and is not fixed for a certain number of comparisons with the reference arm.

3. As already mentioned in Section 2.2, it is not obvious how to define optimality of an arm in general. We ignore this problem for the time being.

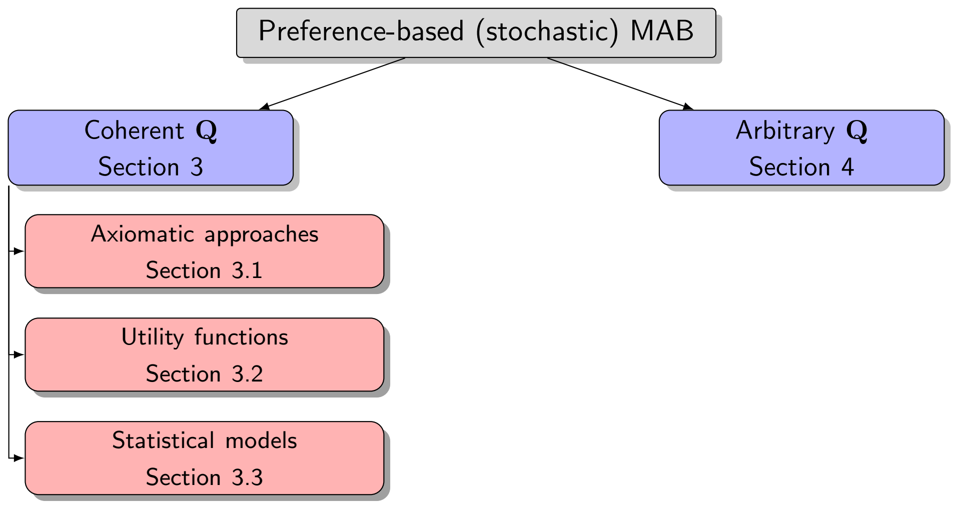

Figure 1: A taxonomy of (stochastic) PB-MAB algorithms.

3. Learning from Coherent Pairwise Comparisons

As explained in Section 2, learning in the PB-MAB setting essentially boils down to estimating the pairwise preference matrix $\mathbf{Q}$, i.e., the pairwise probabilities $q_{i,j}$. Usually, however, the target of the agent's prediction is not the relation $\mathbf{Q}$ itself, but the best arm or, more generally, a ranking $\succ$ of all arms $\mathcal{A}$. As discussed in Section 2.2, the target might not be well-defined for a given preference relation $\mathbf{Q}$, so that information about the latter may not be indicative of the former. Hence, the pairwise probabilities $q_{i,j}$ should be sufficiently coherent, so as to allow the learner to approximate and eventually identify the target (at least in the limit when the sample size grows to infinity).

Different consistency or regularity assumptions on the pairwise probabilities $\mathbf{Q}$ have been proposed in the literature. Needless to say, these assumptions have a major impact on how PB-MAB problems are tackled algorithmically. In this section and the two following ones, we provide an overview of approaches to such problems, categorized according to these assumptions (cf. Figure 1).

3.1 Axiomatic Approaches

We begin this section by collecting various assumptions on pairwise preferences that can be found in the literature. As will be seen later on, by exploiting the (preference) structure imposed by these assumptions, the development of efficient algorithms will become possible.

- *Low noise model*: $\Delta_{i,j} \neq 0$ for all $i \neq j$, and if $\Delta_{i,j} > 0$, then $\sum_{k=1}^{K} \Delta_{i,k} > \sum_{k=1}^{K} \Delta_{j,k}$.

- *Total order over arms*: There is a total order $\succ$ on $\mathcal{A}$, such that $a_i \succ a_j$ implies $\Delta_{i,j} > 0$. The existence of a total order over arms is closely related to different regularity assumptions for triplet of arms, including a stochastic version of the triangle inequality or relaxed notions of transitivity (Haddenhorst et al., 2020):

- *Strong stochastic transitivity* (SST): The inequality $\Delta_{i,k} \geq \max\{\Delta_{i,j}, \Delta_{j,k}\}$ holds for all pairwise distinct $i, j, k \in [K]$ such that $\Delta_{i,j} \geq 0$ and $\Delta_{j,k} \geq 0$.
- *γ-relaxed stochastic transitivity* ($\gamma -$ RST): For $\gamma \in (0, 1)$ and all pairwise distinct $i, j, k \in [K]$, such that $\Delta_{i,j} \geq 0$ and $\Delta_{j,k} \geq 0$, the inequality $\Delta_{i,k} \geq \gamma \max\{\Delta_{i,j}, \Delta_{j,k}\}$ holds.
- *Moderate stochastic transitivity* (MST): The calibrated pairwise probabilities satisfy $\Delta_{i,k} \geq \min\{\Delta_{i,j}, \Delta_{j,k}\}$ for all pairwise distinct $i, j, k \in [K]$ such that $\Delta_{i,j} \geq 0$ and $\Delta_{j,k} \geq 0$.
- *Weak stochastic transitivity* (WST): For any triplet of arms $a_i, a_j, a_k \in \mathcal{A}$, $\Delta_{i,j} \geq 0$ and $\Delta_{j,k} \geq 0$ implies $\Delta_{i,k} \geq 0$.
- *Stochastic triangle inequality* (STI): Given a total order over arms, then for any triplet of arms such that $a_i \succ a_j \succ a_k$, it holds that $\Delta_{i,k} \leq \Delta_{i,j} + \Delta_{j,k}$.

- *General identifiability assumption*: There exists an arm $i^* \in [K]$ such that for any $j \in [K] \backslash \{i^*\}$ it holds that $\min_{k \in [K]} \Delta_{i^*,k} - \Delta_{j,k} > 0$.
- *No ties*: The preference relation $\mathbf{Q}$ is said to have no ties, if $\Delta_{i,j} \neq 0$ for all pairs of distinct arms (a_i, a_j).
- *Existence of a Condorcet winner*: An arm a_i is considered a Condorcet winner if $\Delta_{i,j} > 0$ for all $j \in [K] \backslash \{i\}$, i.e., if it beats all other arms in a pairwise comparison.

Note that WST can also be formulated (as done by some authors) as follows: There is a ranking $\succ$ on $\mathcal{A}$, such that $\Delta_{i,j} \geq 0$ whenever $a_i \succ a_j$. Quite naturally, WST is a necessary and sufficient condition for the existence of a complete ranking of all arms, which is consistent with all pairwise preferences in the sense just mentioned.

In Figure 2, we give an overview of the relationships between the different assumptions, where an arrow represents that one condition implies another one. The implications are quite straightforward to prove, so that in the following we merely provide counterexamples showing why the reversed implications do not hold. However, it is worth noting that the low noise model assumption does not imply any of the triplet conditions within the total order assumption, such as SST or STI. This is illustrated in Figure 2 by the angular rectangle around the triplet conditions. We start with the assumptions in the rounded rectangles in Figure 2:

- Total order over arms $\not\Rightarrow$ low noise model: Let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 1 & 1 & 1 \\ 0 & 0.5 & 0.6 & 0.6 \\ 0 & 0.4 & 0.5 & 1 \\ 0 & 0.4 & 0 & 0.5 \end{pmatrix}.$$

  Then, $a_1 \succ a_2 \succ a_3 \succ a_4$ is the total order for this preference relation matrix. However, it holds that $\Delta_{2,3} > 0$ but $\sum_{k=1}^{4} \Delta_{2,k} = -0.3 < -0.1 = \sum_{k=1}^{4} \Delta_{3,k}$.

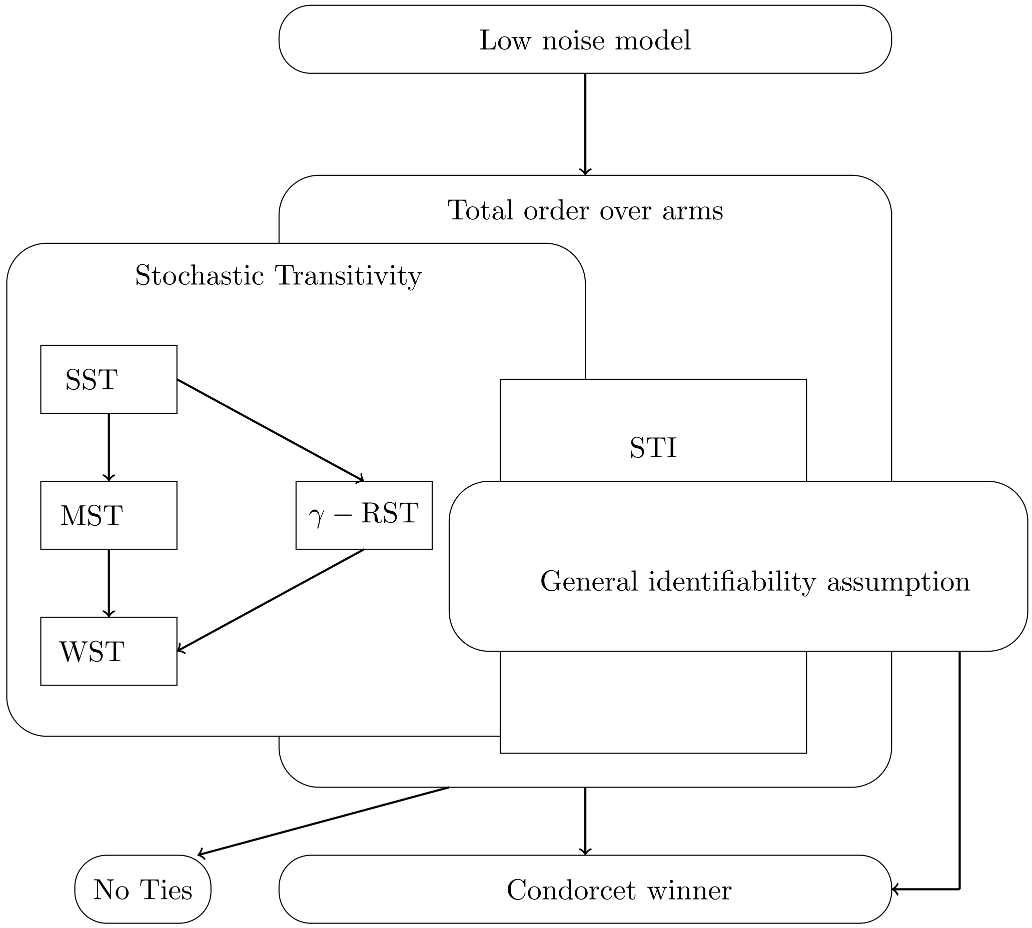

Figure 2: Relationship between the different axiomatic approaches.

- General identifiability assumption $\not\Longrightarrow$ total order over arms $\vee$ SST: Let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 1 & 1 & 1 \\ 0 & 0.5 & 0.9 & 0.1 \\ 0 & 0.1 & 0.5 & 0.9 \\ 0 & 0.9 & 0.1 & 0.5 \end{pmatrix}.$$

  Then, for $i^* = 1$, we have $\max_{j \in [K] \setminus \{1\}} \min_{k \in [K]} \Delta_{i^*,k} - \Delta_{j,k} = 0.1 > 0$, but since $\Delta_{2,3} > 0$, $\Delta_{3,4} > 0$, $\Delta_{2,4} < 0$, there exists no total order for $\mathbf{Q}$. In addition, SST does not hold because of $\Delta_{2,4} = -0.4 < 0.4 = \max\{\Delta_{2,3}, \Delta_{3,4}\}$.

- STI $\not\Longrightarrow$ general identifiability assumption: Let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0.6 & 0.6 \\ 0.4 & 0.5 & 1 \\ 0.4 & 0 & 0.5 \end{pmatrix}.$$

Then, there exists a total order $a_1 \succ a_2 \succ a_3$ and due to $\Delta_{1,3} = 0.1 \leq 0.6 = \Delta_{1,2}+\Delta_{2,3}$ the stochastic triangle inequality is fulfilled. However, it holds that

$$\min_{k=1,2,3} \Delta_{1,k} - \Delta_{2,k} = -0.4 < 0, \quad \min_{k=1,2,3} \Delta_{2,k} - \Delta_{1,k} = -0.1 < 0$$
$$\min_{k=1,2,3} \Delta_{3,k} - \Delta_{1,k} = -0.6 < 0,$$

so that the general identifiability assumption does not hold.

- Total order over arms $\not\Rightarrow$ SST : Consider the latter preference relation $\mathbf{Q}$. Then, there exists a total order of arms but SST does not hold, due to $\Delta_{1,3} = 0.1 < 0.5 = \max\{\Delta_{1,2}, \Delta_{2,3}\}$.

- WST $\not\Rightarrow$ total order over arms $\vee$ general identifiability assumption: Consider the preference relation $\mathbf{Q}$ with all entries set to 0.5, then neither a total order of arms exists nor the general identifiability assumption holds, although WST is fulfilled.

- No ties $\not\Rightarrow$ existence of a Condorcet winner: Let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0.6 & 0.4 \\ 0.4 & 0.5 & 0.6 \\ 0.6 & 0.4 & 0.5 \end{pmatrix}.$$

Then $\mathbf{Q}$ has no ties, but there exists no Condorcet winner, as each arm is beaten by one other arm.

We proceed with the non-obvious transitivity assumptions within the stochastic transitivity properties, namely the relation between MST and RST:

- MST $\not\Rightarrow$ $\gamma -$ RST: Let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0.5+z & 0.5+x \\ 0.5-z & 0.5 & 0.5+y \\ 0.5-x & 0.5-y & 0.5 \end{pmatrix},$$

where $x, y, z \in (0, 1/2)$ are such that $z \leq \min\{x, y\}$ and $x/y < \gamma$. With this, it holds that $a_1 \succ a_2 \succ a_3$ is a total order for $\mathbf{Q}$. Moreover, MST holds because of $\Delta_{1,3} = x \geq z = \min\{y, z\} = \min\{\Delta_{1,2}, \Delta_{2,3}\}$. Yet, $\Delta_{1,3} = x < \gamma y = \gamma \max\{\Delta_{1,2}, \Delta_{2,3}\}$, which is equivalent to $x/y < \gamma$.

- $\gamma -$ RST $\not\Rightarrow$ MST: For any $\gamma \in (0, 1)$, there exists some $\delta > 1$ such that $\gamma \leq 1/\delta$. Fix one such δ, then for any $x \in (0, 1/(2\delta))$, the preference relation given by

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0.5+\delta x & 0.5+x \\ 0.5-\delta x & 0.5 & 0.5+\delta x \\ 0.5-x & 0.5-\delta x & 0.5 \end{pmatrix},$$

implies a total order of arms $a_1 \succ a_2 \succ a_3$ and $\gamma -$ RST, since $\Delta_{1,3} = x \geq \gamma\delta x = \gamma \max\{\Delta_{1,2}, \Delta_{2,3}\}$. On the other hand, MST is not fulfilled due to $\Delta_{1,3} = x < \delta x = \min\{\Delta_{1,2}, \Delta_{2,3}\}$.

Finally, the stochastic triangle inequality is not implied by any transitivity assumptions stronger than WST or vice versa:

- SST $\not\Rightarrow$ STI: For some $\varepsilon \in (0, 1/6)$ let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0.5+\varepsilon & 0.5+3\varepsilon \\ 0.5-\varepsilon & 0.5 & 0.5+\varepsilon \\ 0.5-3\varepsilon & 0.5-\varepsilon & 0.5 \end{pmatrix}.$$

  A total order for $\mathbf{Q}$ exists, namely $a_1 \succ a_2 \succ a_3$, and SST is fulfilled, as $\Delta_{1,3} = 3\varepsilon \geq \varepsilon = \max\{\Delta_{1,2}, \Delta_{2,3}\}$. However, $\Delta_{1,3} = 3\varepsilon > 2\varepsilon = \Delta_{1,2} + \Delta_{2,3}$, so that STI is violated.

- STI $\not\Rightarrow$ MST: For some $\varepsilon \in (0, 1/2)$ let

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 1 & 0.5+\varepsilon \\ 0 & 0.5 & 1 \\ 0.5-\varepsilon & 0 & 0.5 \end{pmatrix}.$$

  Then, $a_1 \succ a_2 \succ a_3$ is a total order for $\mathbf{Q}$, and STI is satisfied, since $\Delta_{1,3} = \varepsilon \leq 1 = \Delta_{1,2} + \Delta_{2,3}$. MST is not present, because of $\Delta_{1,3} = \varepsilon < 0.5 = \min\{\Delta_{1,2}, \Delta_{2,3}\}$

- STI $\not\Rightarrow$ $\gamma-$RST: Consider the preference relation as just defined and set $\varepsilon = \gamma/4$. Then, $\gamma-$RST does not hold due to $\Delta_{1,3} = \gamma/4 < \gamma/2 = \gamma \max\{\Delta_{1,2}, \Delta_{2,3}\}$.

In the following, we review all known methods of the preference-based multi-armed bandit literature for the axiomatic approaches. In particular, the underlying assumptions, the considered goals, the theoretical guarantees (if known), as well as the accompanying novelties of the corresponding methods are concisely described and discussed. Note that the order in which the algorithms are discussed is primarily according to the underlying target and secondarily according to their algorithmic ideas. Finally, Tables 1, 2 and 3 provide a concise overview of the existing methods for the tasks of regret minimization, (ϵ, δ)-PAC learning as well as minimization of the exact sample complexity.

3.1.1 Interleaved Filtering

Assuming a total order over arms, strong stochastic transitivity, and the stochastic triangle inequality, Yue et al. (2012) propose an explore-then-exploit algorithm. The exploration step consists of a simple sequential elimination strategy, called Interleaved Filtering (IF), which identifies the best arm with probability at least $1-\delta$. The IF algorithm successively selects an arm which is compared to other arms in a one-versus-all manner. More specifically, the currently selected arm a_i is compared to the rest of the active (not yet eliminated) arms. If an arm a_j beats a_i, that is, $\widehat{q}_{i,j} + c_{i,j} < 1/2$, then a_i is eliminated, and a_j is compared to the rest of the (active) arms, again in a one-versus-all manner. In addition, a simple pruning technique can be applied: if $\widehat{q}_{i,j} - c_{i,j} > 1/2$ for an arm a_j at any time, then a_j can be eliminated, as it cannot be the best arm anymore (with high probability) due to the underlying transitivity assumption. After the exploration step, the

4. This assumption is simplified here for sake of convenience. Refer to Section 3.1.4 for the exact formulation of the assumption.

Algorithm	Algorithm class(es)	Assumption(s)	Target(s) and goal(s) of learner	Theoretical guarantee(s)
Interleaved Filtering (Section 3.1.1)	Generalization (Explore-then-exploit), Challenge	A priori known time horizon T, Total order over arms, SST and STI	Expected regret minimization	$\mathcal{O}\left(\frac{K \log T}{\min_{j \neq i^*} \Delta_{i^*,j}}\right)$
Beat the Mean (Section 3.1.2)	Generalization (Explore-then-exploit), Challenge	A priori known time horizon T, Total order over arms, RST and STI	High probability regret minimization	$\mathcal{O}\left(\frac{K \log T}{\gamma^7 \min_{j \neq i^*} \Delta_{i^*,j}}\right)$
Relative Upper Confidence Bound (Section 3.1.3)	Generalization (UCB)	Existence of a Condorcet winner	Expected and high probability regret minimization	$\mathcal{O}\left(K^2 + \sum_{i \neq i^*} \frac{\log T}{\Delta^2_{i^*,i}}\right)$
MergeRUCB (Section 3.1.4)	Generalization (UCB), tournament	No ties[4] and existence of a Condorcet winner	High probability regret minimization	$\mathcal{O}\left(\frac{K \log(T)}{\min_{i,j:\, q_{i,j} \neq 1/2} \Delta^2_{i,j}}\right)$
MergeDTS (Section 3.1.5)	Generalization (Thompson-Sampling), tournament	A priori known time horizon T and same assumptions as for MergeRUCB	High probability regret minimization	$\mathcal{O}\left(\frac{K \log(T)}{\min_{i,j:\, q_{i,j} \neq 1/2} \Delta^2_{i,j}}\right)$
Relative Confidence Sampling (Section 3.1.6)	Generalization (UCB and Thompson sampling), tournament	Existence of a Condorcet winner	Expected and high probability regret minimization	No theoretical guarantees
Relative Minimum Empirical Divergence (Section 3.1.7)	Generalization (DMED)	Existence of a Condorcet winner	Expected regret minimization	$\mathcal{O}\left(\sum_{i \neq i^*} \frac{\Delta_{i^*,i} \log T}{\mathrm{KL}(q_{i,i^*}, 1/2)} + K^{2+\varepsilon}\right)$
Winner Stays (Section 3.1.8)	Challenge	No ties and either 1. Existence of a Condorcet winner, or 2. Total order over arms	Expected regret minimization with (a) weak regret (b) strong regret	1.(a): $\mathcal{O}\left(\frac{K^2}{\min_{i,j} \lvert\Delta_{i,j}\rvert^3}\right)$ 1.(b): $\mathcal{O}\left(\frac{K^2}{\min_{i,j} \Delta^2_{i,j}} + \frac{K \log T}{\min_{i,j} \lvert\Delta_{i,j}\rvert}\right)$ 2.(a): $\mathcal{O}\left(\frac{K \log K}{\min_{i,j} \Delta^6_{i,j}}\right)$ 2.(b): $\mathcal{O}\left(\frac{K \log K}{\min_{i,j} \Delta^6_{i,j}} + \frac{K \log T}{\min_{i,j} \lvert\Delta_{i,j}\rvert}\right)$
Beat the Winner (Section 3.1.9)	Challenge	Existence of a Condorcet winner	Expected regret minimization with weak regret	$\mathcal{O}\left(K^2 + \frac{K}{(1-\exp(-2\min_{j \neq i^*} \lvert\Delta_j\rvert^2))^2}\right)$

Table 1: Algorithms for the regret minimization task under axiomatic approaches. The index i^* is representing the best arm and $\mathrm{KL}(p, q)$ is the Kullback-Leibler divergence of Bernoulli random variables with parameters p and q.

Algorithm	Algorithm class(es)	Assumption(s)	Target(s) and goal(s) of learner	Theoretical guarantee(s)
Beat the Mean (Section 3.1.2)	Challenge	Total order over arms, γ-RST and STI	(ϵ, δ)-PAC for best arm	$\mathcal{O}\left(\frac{K}{\gamma^6 \epsilon^2} \log \frac{K}{\epsilon\delta}\right)$
Knockout Tournaments (Section 3.1.10)	Tournament	Total order over arms, SST, STI, γ-RST	(ϵ, δ)-PAC for best arm	$\mathcal{O}\left(\frac{K}{\gamma^4 \epsilon^2} \log \frac{1}{\delta}\right)$
Sequential Elimination (Section 3.1.11)	Challenge	Total order over arms, SST	(ϵ, δ)-PAC for best arm	$\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\delta}\right)$
Opt-Max (Section 3.1.12)	Tournament	Total order over arms, MST	(ϵ, δ)-PAC for best arm	If $\delta \geq \min(1/K, \exp(-K^{1/4}))$: $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\delta}\right)$
Binary-Search-Ranking (Section 3.1.13)	Tournament	Total order over arms, SST, STI	(ϵ, δ)-PAC for ranking	If $\delta \geq 1/K : \mathcal{O}\left(\frac{K \log K}{\epsilon^2}\right)$
Noisy Quick Select (Section 3.1.14)	Noisy-sorting, Tournament	Total order over arms, SST, STI	(ϵ,δ)-PAC for Top-k identification (for $1 \leq k \leq K/2$)	$\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{k}{\delta}\right)$
Approx-Prob (Section 3.1.15)	Tournament	Total order over arms, SST and STI	(ϵ, δ)-PAC for estimation of $\mathbf{Q}$	If $\delta \geq 1/K$: $\mathcal{O}\left(\frac{K \min(K, 1/\epsilon) \log(K)}{\epsilon^2}\right)$

Table 2: Algorithms for (ϵ, δ)-PAC tasks under axiomatic approaches.

exploitation step simply takes the best arm $a_{\widehat{i}^*}$ found by IF and repeatedly compares $a_{\widehat{i}^*}$ to itself.

The authors analyze the expected regret achieved by IF. Assuming the horizon T to be finite and known in advance, they show that IF incurs an expected regret of order $\mathcal{O}\left(\frac{K}{\min_{j \neq i^*} \Delta_{i^*,j}} \log T\right)$, which is shown to be the lower bound in this case as well.

3.1.2 Beat the Mean

In a subsequent work, Yue and Joachims (2011) relax the strong stochastic transitivity property and only require relaxed stochastic transitivity for the pairwise probabilities. Further, both the relaxed stochastic transitivity and the stochastic triangle inequality are required to hold only relative to the best arm, i.e., only for triplets where i is the index of the best arm a_{i^*}.

With these relaxed properties, Yue and Joachims (2011) propose a preference-based online learning algorithm called Beat-The-Mean (BTM), which is an elimination strategy resembling IF. However, while IF compares a single arm to the rest of the (active) arms in a one-versus-all manner, BTM selects an arm with the fewest comparisons so far and pairs it with a randomly chosen arm from the set of active arms (using the uniform distribution). Based on the outcomes of the pairwise comparisons, a score b_i is assigned to each active arm a_i, which is an empirical estimate of the probability that a_i is winning in a pairwise comparison (not taking into account which arm it was compared to). The idea is that comparing an arm a_i to the "mean" arm, which beats half of the arms, is equivalent to comparing a_i to an arm randomly selected from the active set. One can deduce a confidence interval for the scores b_i, which allows for deciding whether the scores for two arms are

Algorithm	**Algorithm class(es)**	**Assumption(s)**	**Target(s) and goal(s) of learner**	**Theoretical guarantee(s)**
Robust Query Selection (Section 3.1.16)	Generalization	Total order over arms and embedding of the arms into a d-dimensional Euclidean space	Exact sample complexity for ranking	$\mathcal{O}\left(\frac{d}{\min_{1\leq i<j\leq K}\Delta_{i,j}^2}\log^2\left(\frac{K}{\delta}\right)\right)$
Verification-based Solution (Section 3.1.17)	Generalization (exploration and verification)	Existence of a Condorcet winner	Exact sample complexity for best arm	$\mathcal{O}\Big(\sum_{i\neq i^*}\min_{j:q_{i,j}<1/2}\frac{\log\left(K/(\delta\Delta_{i,j}^2)\right)}{\Delta_{i,j}^2}\Big)$ $+\tilde{\mathcal{O}}\Big(\sum_{i\neq i^*}\left(\Delta_{i^*,i}^{-2}+\sum_{j\neq i}\Delta_{i,j}^{-2}\right)\Big)$
Parallel Selection and Partition (Section 3.1.18)	Noisy-sorting	Total order over arms	Exact sample/round complexity for top-k identification	$\mathcal{O}\left(\frac{K}{\min_{1\leq i<j\leq K}\Delta_{i,j}^2}\log(K)\right)$
Single Elimination Tournament (Section 3.1.19)	Noisy-sorting, tournament	Total order over arms	Exact sample complexity for 1. Best arm 2. Top-k ranking 3. Top-k identification	1. $\mathcal{O}\left(\frac{K\log\log(K)}{\min_{j\neq i^*}\Delta_{i^*,j}^2}\right)$ 2. $\mathcal{O}\left(\frac{(K+k\log k)\max\{\log k,\log\log K\}}{\min_{i\in[k]}\min_{j:j\geq i}\Delta_{i,j}^2}\right)$ 3. $\mathcal{O}\left(\frac{(K+k\log k)\max\{\log k,\log\log K\}}{\Delta_{(k),(k+1)}^2}\right)$
Sequential Elimination Exact Selection (Section 3.1.20)	Noisy-sorting, challenge	Total order over arms, SST, STI	Exact sample complexity for 1. Best arm 2. Top-k identification	1. $\mathcal{O}\left(\sum_{i\in[K]}\frac{\log 1/\delta+\log\log 1/\Delta_{i,(k)}}{\Delta_{i,(k)}^2}\right)$ 2. $\mathcal{O}\left(\sum_{i\in[K]}\frac{\log K/\delta+\log\log 1/\Delta_{i,(k)}}{\Delta_{i,(k)}^2}\right)$
Iterative-Insertion-Ranking (Section 3.1.21)	Noisy-sorting	Total order over arms, SST	Exact sample complexity for ranking	$\mathcal{O}\left(\sum_{i\in[K]}\frac{(\log\log(\min_{j\neq i}\Delta_{i,j}^{-1})+\log(K/\delta))}{\min_{j\neq i}\Delta_{i,j}^2}\right)$

Table 3: Algorithms for exact sample complexity tasks (corresponding to the $(0,\delta)$-PAC learning scenario) under axiomatic approaches. The definitions of $\Delta_{i,(k)}$ and $\Delta_{(k),(k+1)}$ can be found in the respective sections.

significantly different. An arm is then eliminated as soon as there is another arm with a significantly higher score.

In the regret analysis of BTM, a high probability bound is provided for a finite time horizon. More precisely, the regret accumulated by BTM is $\mathcal{O}\left(\frac{K}{\gamma^7 \min_{j \neq i^*} \Delta_{i^*,j}} \log T\right)$ with high probability. This result is stronger than the one proven for IF, in which only the expected regret is upper-bounded. Moreover, this high probability regret bound matches with the expected regret bound in the case $\gamma = 1$ (strong stochastic transitivity).

The authors also analyze the BTM algorithm in a PAC setting for finding the best arm, i.e., for any given $\epsilon, \delta > 0$, the algorithm should find an ϵ-optimal arm (cf. Section 2.2.5) with probability at least $1 - \delta$, while keeping the number of overall duels as low as possible. It is shown that BTM is an (ϵ, δ)-PAC preference-based learner (by setting its input parameters appropriately) with a sample complexity of $\mathcal{O}(\frac{K}{\gamma^6 \epsilon^2} \log \frac{KN}{\delta})$ if N is large enough, that is, N is the smallest positive integer for which $N = \left\lceil \frac{36}{\gamma^6 \epsilon^2} \log \frac{K^3 N}{\delta} \right\rceil$ holds. One may simplify this bound by noting that $N < N' = \left\lceil \frac{864}{\gamma^6 \epsilon^2} \log \frac{K}{\delta} \right\rceil$. Therefore, the sample complexity of BTM is $\mathcal{O}\left(\frac{K}{\gamma^6 \epsilon^2} \log \frac{K \log(K/\delta)}{\delta \, \gamma \, \epsilon}\right)$.

3.1.3 Relative Upper Confidence Bound

In a work by Zoghi et al. (2014b), the well-known UCB algorithm (Auer et al., 2002a) is adapted from the value-based to the preference-based MAP setup in order derive an algorithm minimizing the cumulative regret (see (2)). One of the main advantages of the proposed algorithm, called RUCB (for Relative UCB), is that only the existence of a Condorcet winner is required. The RUCB algorithm is based on the "optimism in the face of uncertainty" principle, which means that the arms to be compared next are selected based on the optimistic estimates of the pairwise probabilities, that is, based on the upper bounds $\widehat{q}_{i,j} + c_{i,j}$ of the confidence intervals. In an iteration step, RUCB selects the set of potential Condorcet winners for which all $\widehat{q}_{i,j} + c_{i,j}$ values are above 1/2, and then selects an arm a_i from this set uniformly at random. Finally, a_i is compared to the arm a_j, where $j = \operatorname{argmax}_{\ell \neq i} \widehat{q}_{i,\ell} + c_{i,\ell}$, that may lead to the smallest regret, taking into account the optimistic estimates.

In the analysis of the RUCB algorithm, horizonless bounds are provided, both for the expected and high probability regret. Thus, unlike the bounds for IF and BTM, these bounds are valid for each time step. Both the expected regret bound and high probability bound of RUCB are

$$\mathcal{O}\Big(K^2 + \sum_{i \neq i^*} \frac{\log T}{\Delta_{i^*,i}^2}\Big).$$

However, while the regret bounds of IF and BTM only depend on $\min_{j \neq i^*} \Delta_{i^*,j}$, the constants are now of different nature, despite being still calculated based on the $\Delta_{i,j}$ values. Therefore, the regret bounds for RUCB are not directly comparable with those given for IF and BTM. Moreover, the regret bound for IF and BTM is derived based on the explore-and-exploit technique, which requires the knowledge of the horizon in advance, whereas regret bounds for RUCB, both high probability and expectation, are finite time bounds that hold for any time step T.

3.1.4 MergeRUCB

Zoghi et al. (2015b) consider the same problem as in the previous section, but with a special focus on learning scenarios in which the number of available arms is large. In order to keep the number of comparisons small, they propose the MergeRUCB algorithm which, using a similar divide-and-conquer strategy as the merge sort algorithm, proceeds by first grouping the arms in batches of a predefined size and then processing them separately before merging them together. In particular, only arms within the same batch can be compared with each other, but not arms in different batches. Due to the stochastic nature of the feedback, the local comparisons within each batch between two arms are run multiple times before eliminating inferior arms based on upper confidence bounds of the preference probabilities. More precisely, an arm is eliminated within one batch if its upper confidence bound on the pairwise winning probability with respect to some arm in the same batch is below 1/2. In each time step, MergeRUCB chooses one batch in a round-robin manner, while the choice for the arms compared within one batch is made by choosing one arm uniformly at random and comparing it with its worst competitor, i.e., the arm having the highest chance of leading to an elimination of the first chosen arm. In light of this, it is ensured that the batch sizes are reduced quickly, and in turn a merge step can be performed early, which happens as soon as the sum of the batch sizes are below a stage-wise geometrically decreasing threshold, while a stage corresponds to the overall number of conducted merge steps. Within one merge step, batches of smaller sizes are grouped together with batches of larger sizes. This entire iterative process will eventually end with one single batch left, consisting of only one arm, which is then guaranteed to be the Condorcet winner with high probability.

For the theoretical analysis of MergeRUCB, it is assumed that for any pair of arms (a_i, a_j) it holds that either their pairwise winning probability is different from 1/2 (i.e., $\Delta_{i,j} \neq 0$), or they are inferior to any other arm a_k in the sense that $\max\{\Delta_{i,k}, \Delta_{j,k}\} < 0$ holds. If the latter property holds for a pair of arms, then this pair is called *uninformative*. The authors assume that at most a third of the arms are uninformative, which guarantees that after a specific number of merge steps at least one arm will be present in the batch in order to eliminate all other arms of that batch. Moreover, these assumptions allow one to derive a high probability bound on the cumulative regret of MergeRUCB, which is of the order $\mathcal{O}\Big(\frac{K \log(T)}{\min_{j \neq i : q_{i,j} \neq 1/2} \Delta_{i,j}^2}\Big)$, thereby eliminating the additive K^2 term in the regret bound of RUCB, as pairwise comparisons are only carried out within the local batches but not "globally" as in RUCB.

3.1.5 MergeDTS

Under the same assumptions as MergeRUCB, Li et al. (2020) propose the Merge Double Thompson Sampling (MergeDTS) algorithm, which improves upon the former by using the Double Thompson Sampling algorithm (cf. Section 4.2.3) in order to choose a pair of arms for the comparison within one batch. It is shown that MergeDTS enjoys the same order as MergeRUCB on its upper bound of the regret with high probability, but in contrast to the latter, needs to know the time horizon beforehand. However, in an extensive experimental study, the authors show that MergeDTS is superior to MergeRUCB and other state of the art dueling bandits algorithms.

3.1.6 Relative Confidence Sampling

Again merely assuming the existence of a Condorcet winner and focusing on the minimization of the cumulative regret, Zoghi et al. (2014a) introduce the relative confidence sampling (RCS) algorithm, which, in addition to the upper confidence bounds of the entries of the preference relation $\mathbf{Q}$ (such as RUCB), maintains a Beta posterior distribution over the entries, respectively. The idea is that both upper confidence bounds and the Beta posterior distributions are used to suggest one arm each for the duel at one iteration step. To this end, a preference relation $\tilde{\mathbf{Q}} \in [0,1]^{K \times K}$ is sampled[5] according to the current Beta posterior distributions. If $\tilde{\mathbf{Q}}$ has a Condorcet winner, then this arm is chosen as the first arm for the duel, otherwise the arm with the fewest picks according to the choice mechanism of the first case is used. As the second arm of the duel, RCS chooses the toughest competitor of the first arm, namely the arm that has the highest (optimistic) chance to confute the Condorcet winner property of the latter (similar as in MergeRUCB) according to the current upper confidence bounds. The authors present experimental results on learning-to-rank data sets, revealing a satisfactory empirical performance of RCS regarding cumulative regret, but do not provide theoretical guarantees for RCS in terms of an upper bound on its cumulative regret.

3.1.7 Relative Minimum Empirical Divergence

Komiyama et al. (2015) assume that the underlying preference matrix has a Condorcet winner, and propose three variants of the relative minimum empirical divergence (RMED) algorithm, which can be interpreted as a dueling bandits variant of the deterministic minimum empirical divergence (DMED) algorithm (Honda and Takemura, 2010) for the value-based MAB problem. More specifically, the algorithm revolves around *the empirical divergence of an arm a_i* at time t defined by

$$I_{a_i}(t) = \sum_{\{a_j : \widehat{q}^t_{i,j} \leq 1/2\}} n^t_{i,j} \ \mathrm{KL}(\widehat{q}^t_{i,j}, 1/2),$$

where $\mathrm{KL}(p,q)$ denotes the Kullback-Leibler divergence of Bernoulli random variables with success probabilities p and q. As the exponential negative empirical divergence of an arm a_i can be interpreted as the likelihood of being the Condorcet winner, one can define the empirical best arm by means of $i^*_t = \mathrm{argmin}_{i \in [K]} I_{a_i}(t)$. Further, the algorithm maintains a set of potentially good arms defined by $C_t = \{i \in [K] \,|\, I_{a_i}(t) - I_{i^*_t}(t) \leq \log(t) + f(K)\}$ for some non-negative function f that does not depend on t. After a variant-specific exploration phase, all three variants of the RMED algorithm are in each time step essentially comparing one specifically chosen arm in C_t (based on some ordering) with either the empirically best arm or, depending on the RMED variant, one specific arm based on the first chosen arm. In particular, the first variant, called RMED1, chooses in case the empirically best arm is empirically not preferred over the first chosen arm, the arm which is empirically preferred the most over the latter. Here, "empirically" means that the choices are based on the current empirical pairwise estimates $(\widehat{q}^t_{i,j})_{1 \leq i,j \leq K}$. For this variant, a bound on its

5. Strictly speaking, the preference relation $\tilde{\mathbf{Q}}$ depends on the iteration step t, as the Beta posterior distributions do. For sake of convenience, we suppress this dependency here in the notation.

expected regret of order $\mathcal{O}\left(\sum_{i \neq i^*} \frac{\log T \, \Delta_{i^*,i}}{\mathrm{KL}(q_{i,i^*},1/2)}\right) + \mathcal{O}(K^{2+\varepsilon})$ is shown, where $\varepsilon > 0$ is a parameter of the algorithm specifying the used function f for the set of potentially good arms. Further, using similar proof techniques as Lai and Robbins (1985) for showing asymptotic lower bounds in the standard MAB problem, they derive an asymptotic lower bound[6] of order $\Omega\left(\sum_{i \neq i^*} \min_{j: q_{i,j} < 1/2} \frac{(\Delta_{i^*,i} + \Delta_{i^*,j}) \log T}{\mathrm{KL}(q_{i,j},1/2)}\right)$ for the regret of any consistent dueling bandits algorithm for preference relations having a Condorcet winner or admitting a total order of the arms. This result reveals that there is a gap between the upper bound of RMED1 and the asymptotic lower bound regarding the constant factors.

In order to obtain an algorithm having a regret bound with the constant factor matching the asymptotic lower bound (i.e., which is asymptotically optimal), they suggest the second variant of the RMED algorithm, called RMED2, which adapts the mechanism of RMED1 for choosing the second arm in order to obtain an estimate of the gap between the constant factor of RMED1 and the asymptotic lower bound of the first chosen arm. Because the theoretical analysis of RMED2 is cumbersome, the third variant of RMED, called RMED2 Fixed Horizon (RMED2FH), is introduced. In contrast to the first two variants, RMED2FH needs to know the time horizon T beforehand, because this is used to derive a "non-exploring" estimate for the gap between the constant factor of RMED1 and the asymptotic lower bound of the first chosen arm, facilitating the theoretical analysis considerably. Under this assumption and the existence of a Condorcet winner, it is shown that RMED2FH enjoys an asymptotically optimal regret upper bound.

3.1.8 Winner Stays

Chen and Frazier (2017) study the dueling bandits problem in the Condorcet winner setting, and consider both extreme cases for the regret, namely the strong regret specified by the instantaneous regret $r_{t,max}$ and the weak regret $r_{t,min}$, which is 0 if either arm pulled is the Condorcet winner. They propose the Winner Stays (WS) algorithm with variations for both kinds of regret. WS for weak regret (WS-W) is a round-based challenge algorithm, where arms are dueled with each other in a batch of duels in each round, and each batch corresponds to a sequence of duels of the same pair consisting of the current champion and a "worthy" challenger. The current champion is the arm that has currently the largest number of overall duels won, while the challenger is the arm that has not yet been considered in the current round and has currently the second largest number of overall duels won (possibly breaking ties). The batch of duels continues until either (i) the challenger manages to win against the current champion so many times that the champion's overall number of duels won is the third highest, or (ii) the champion wins against the challenger so many times that the challenger's overall number of duels won is the third highest. In case of the first event, the challenger becomes the current champion, while in the second event, the champion (winner) stays. In both cases, the defeated arm is not considered anymore in the current round and the next round starts after each available arm has been considered at least once in a batch of duels.

If the underlying instantaneous regret is the strong regret, the authors propose the WS for strong regret (WS-S), which considers separate exploration and exploitation phases in

6. This asymptotic lower bound $\Omega(C \cdot f(T))$ is to be understood as $\liminf_{T \to \infty} \frac{\mathbf{E}[R^T]}{f(T)} \geq C$.

epochs. Within each epoch $e \in \mathbb{N}$, first the eth round of the WS-W algorithm is conducted for the exploration phase, resulting in a current champion that is then dueled against itself (cf. "fully commitment" in Section 2.3.1) in the exploitation phase for the purpose of keeping the cumulative strong regret low. In light of this, the length of an exploitation phase is exponentially increasing with the number of epochs passed.

Assuming that there are no ties in the underlying preference relation $\mathbf{Q}$, it is proven that unlike all regret bounds for cumulative average regret, the WS-W algorithm has expected cumulative weak regret that is constant in time. In particular, it is shown that WS-W has an expected cumulative weak regret bound of order $\mathcal{O}\left(\frac{K^2}{\min_{i,j}|\Delta_{i,j}|^3}\right)$ under the assumption of an existing Condorcet winner, and of order $\mathcal{O}\left(\frac{K\log K}{\min_{i,j}\Delta_{i,j}^6}\right)$ under the assumption of an existing total order of arms. Further, it is proved that WS-S enjoys an expected cumulative strong regret bound of order $\mathcal{O}\left(\frac{K^2}{\min_{i,j}\Delta_{i,j}^2} + \frac{K\log T}{\min_{i,j}|\Delta_{i,j}|}\right)$ in the Condorcet winner setting, and of order $\mathcal{O}\left(\frac{K\log K}{\min_{i,j}\Delta_{i,j}^6} + \frac{K\log T}{\min_{i,j}|\Delta_{i,j}|}\right)$ under the assumption of an existing total order of arms. Both bounds are optimal regarding their dependence on the time horizon T, but are not optimal regarding the dependence on the calibrated preference probabilities. The proof for WS-W is revised by Peköz et al. (2020), who show that WS-W incurs in fact an expected regret upper bound of $\mathcal{O}\left(\frac{K^2}{\min_{j\neq i^*}|\Delta_{i^*,j}|^2}\right)$ under the assumption of an existing Condorcet winner, but without assuming that there are no ties in the underlying preference relation $\mathbf{Q}$.

It is worth noting that the analysis of the WS algorithms is in some sense unique, as the Gambler's ruin problem is used to upper bound the number of pulls of sub-optimal arms, whereas all regret minimizing algorithms reviewed so far make use of the Chernoff bound in some way.

3.1.9 Beat the Winner

Having the same target as WS-W, Peköz et al. (2020) propose the Beat the Winner (BTW) algorithm, which is a round-based challenge algorithm based on a queue structure of the arms. Here, the first arm in the queue is the current champion while the second is its challenger. Both arms are compared in round $l \in \mathbb{N}$ as long as one of the two arms has won exactly l many times, whereupon the defeated arm is set to the end of the queue and the winner is queued to the front (if necessary). It is shown that BTW enjoys an upper bound on its expected weak regret of $\mathcal{O}\left(K^2 + \frac{K}{(1-\exp(-2\min_{j\neq i^*}|\Delta_j|^2))^2}\right)$, where the additive linear (in K) term might dominate the quadratic term for a small total number of arms K.

As BTW does not consider the history of the past duels in an explicit way, the authors suggest the Modified Beat the Winner (MBTW) algorithm, which improves upon BTW by assigning scores to arms based on their history of wins and chooses the challenger in each round in a random manner based on their relative scores. The initial scores of the arms are set to 1 and increased/decreased by one for the winner/loser of a respective round, which, however, cannot fall below a score of one. Although MBTW is not theoretically analyzed, it is shown to have a satisfactory empirical performance compared to BTW as well as WS-W for the task of expected weak regret minimization.

3.1.10 Knockout Tournaments

Assuming a total order over the arms, γ-relaxed stochastic transitivity as well as stochastic triangle inequality, Falahatgar et al. (2017b) consider the goals of finding the best arm as well as the best ranking (cf. Section 3.1.13) in the (ϵ, δ)-PAC setting. More specifically, for any given $\epsilon, \delta > 0$, the algorithm for finding the best arm must output an ϵ-optimal arm (cf. Section 2.2.5) with probability at least $1 - \delta$.

For this purpose, they propose the Knockout algorithm, which proceeds in rounds, each of which corresponds to a knockout tournament with the goal of successively eliminating half of all currently remaining arms until only one single arm is left, which is then the suggested candidate for the best arm, i.e., an ϵ-optimal arm. At the beginning of each round, all not yet eliminated arms are divided into pairs in a random manner. Then, all these pairs of arms are successively dueled with each other until either one is confident enough which arm is superior resp. inferior, or a certain (round-dependent) number of duels has been reached, whereupon the arm with the larger number of duels won proceeds to the next round, while the other one is eliminated. Thanks to the regularity assumptions made on $\mathbf{Q}$, it is ensured by choosing a suitable maximal limit on the (round-dependent) number of duels that the highest ranked arm at the beginning of a round and the highest ranked arm at the end of a round are close in terms of their calibrated preference probabilities, which can be upper bounded by a term that is linear in the used approximation quality of a round. To maintain the overall confidence level δ and the approximation quality ϵ of the entire procedure, and at the same time prevent a larger sample complexity through a rough Bonferroni correction, both the round-wise confidence level and the round-wise approximation quality are geometrically progressing. In this way, it can be shown that Knockout is an (ϵ, δ)-PAC algorithm for finding the best arm with a sample complexity of $\mathcal{O}\left(\frac{K}{\gamma^4 \epsilon^2} \log \frac{1}{\delta}\right)$. This improves upon the sample complexity shown for BTM for the same setting (cf. Section 3.1.2) and matches the lower bound for $\gamma = 1$ (i.e., strong stochastic transitivity), which can be derived by Feige et al. (1994).

3.1.11 Sequential Elimination

Seeking the same goals as Falahatgar et al. (2017b), but this time only requiring a total order over the arms and strong stochastic transitivity (no stochastic triangle inequality), Falahatgar et al. (2017a) present the Seq-Eliminate algorithm in an (ϵ, δ)-PAC setting for finding the best arm, which uses $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\delta}\right)$ comparisons. The Seq-Eliminate algorithm is a challenge algorithm (cf. Section 2.4.5), which does not use an arm for dueling if this arm has been defeated once. In particular, Seq-Eliminate starts by selecting a current "champion" at random, and keeps dueling it with another random arm (challenger) until the more preferred arm of the two is determined with a certain confidence. It then proceeds to the next competition stage, after setting the winner from the last stage as the new champion and eliminating the loser. The algorithm stops as soon as only a single arm remains. Unlike Knockout, the confidence levels within each competition stage are designed in a more adaptive way, which ensures that with this elimination procedure the overall confidence level δ as well as the approximation quality ϵ are maintained.

Using a random choice to determine the first champion, Seq-Eliminate has in fact a sample complexity of $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{K}{\delta}\right)$, which is order optimal in the high confidence regime, i.e., if $\delta \leq 1/K$, but not if $\delta > 1/K$. In order to deal with confidence levels of the latter kind, the authors propose to find first a good initial champion, say $\tilde{a}$, by using Seq-Eliminate on a randomly sampled smaller subset $\mathcal{A}$ of a specific size. Next, $\tilde{a}$ is used to obtain an auxiliary and potentially smaller subset of all arms, say $\tilde{A}$, by dueling it with all arms multiple times in a competition stage manner (with specifically chosen confidence levels and approximation quality) and include an arm in $\tilde{A}$ only if it has managed to withstand $\tilde{a}$ in all stages. Finally, each arm in $\tilde{A}$ is dueled once again in a competition stage manner against $\tilde{a}$ until either $\tilde{a}$ is winning against each arm in $\tilde{A}$, making $\tilde{a}$ the final output, or $\tilde{a}$ is inferior to one arm in $\tilde{A}$, whereupon Seq-Eliminate is used on $\tilde{A}$ to obtain the final output. With this modification of Seq-Eliminate, the authors show that one obtains an (ϵ, δ)-PAC algorithm for finding the best arm, which uses $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\delta}\right)$ comparisons for $\delta > 1/K$.

3.1.12 Opt-Max

Replacing the strong stochastic transitivity by the moderate stochastic transitivity assumption, Falahatgar et al. (2018) study the problem of best arm identification in an (ϵ, δ)-PAC setting. They present the Opt-Max algorithm, which makes heavily use of a subroutine, called Soft-Seq-Elim. The latter essentially operates as Seq-Eliminate, but can also refrain from eliminating one arm after a sequence of duels in a stage, if no clear winner based on a specific confidence level can be declared. Thus, Soft-Seq-Elim terminates if the current champion has not changed after dueling it with all active arms.

The guarantees and the sample complexity of Soft-Seq-Elim critically depend on the number of changes of the champion: Although the worst-case sample complexity of the algorithm is quadratic, it runs fast (close to linear) and tends to yield correct answers when the number of required changes is small. In other words, Soft-Seq-Elim strongly benefits from the choice of a "good" initial champion (ideally the best arm) in the beginning. The Opt-Max algorithm consists of three variants of Soft-Seq-Elim, each of them tailored to a certain range of the confidence level δ by adapting it essentially in a similar manner as Seq-Eliminate for the low confidence regime (cf. Section 3.1.11).

It is shown that Opt-Max is an (ϵ, δ)-PAC algorithm for finding the best arm having a sample complexity of order $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\delta}\right)$, at least if $\delta \geq \min(1/K, \exp(-K^{1/4}))$. In light of the findings by Feige et al. (1994), the order of Opt-Max' sample complexity is optimal (cf. also Section 3.1.10). Finally, the authors show that any algorithm for finding the best arm in the PAC-setting requires in the worst-case scenario a number of comparisons that scales with K^2 if merely weak stochastic transitivity is assumed.

3.1.13 Binary-Search-Ranking

For the ranking problem in an (ϵ, δ)-PAC setting, the authors of the Knockout algorithm (Falahatgar et al. (2017b)) also propose the Binary-Search-Ranking algorithm, assuming that the underlying preference relation $\mathbf{Q}$ admits a total order over the arms and satisfies strong stochastic transitivity as well as the stochastic triangle inequality.

This algorithm consists of three major steps. In the first step, it randomly selects a set of arms of size $\frac{K}{(\log K)^x}$, called anchors, and ranks them using a procedure called Rank-x—an (ϵ, δ)-PAC ranking algorithm, which for any $x > 1$, uses $\mathcal{O}\left(\frac{K}{\epsilon^2} (\log K)^x \log \frac{K}{\delta}\right)$ comparisons, while at the same time creating $\frac{K}{(\log K)^x} - 1$ bins between each two successive anchors. Then, in the second step, a random walk on a binary search tree is used to assign each arm to a bin. Finally, in the last step, the output ranking is produced. To this end, the arms that are close to an anchor are ranked close to it, while arms that are distant from two successive anchors are ranked using Rank-x.

In a subsequent work, Falahatgar et al. (2018) propose an improvement of the latter, which gets rid of superfluous logarithmic terms in the sample complexity. This improvement is achieved by modifying the components of the algorithm as follows. Each component is called a first time with the (high probability) guarantee of a correct output of $1 - 1/\log(K)$ instead of $1 - 1/K^5$, resulting in a smaller number of comparisons. Then, the output of the respective component is checked, for which a small complexity can be shown. Finally, if the output is incorrect, the corresponding component is run again, but this time with a $1 - 1/K^5$ guarantee for its correctness.

It is shown that the (enhanced) Binary-Search-Ranking algorithm is an (ϵ, δ)-PAC algorithm for the ranking problem with sample complexity $\mathcal{O}\left(\frac{K \log K}{\epsilon^2}\right)$ if δ is set to $\frac{1}{K}$. Thus, the leading factor of the sample complexity of finding a nearly best arm differs from finding a nearly best ranking by a logarithmic factor. This was to be expected, and simply reflects the difference in the worst-case complexity for finding the largest element in an array and sorting an array using an efficient sorting strategy.

Falahatgar et al. (2017b) derive a lower bound on the sample complexity of $\Omega\left(\frac{K}{\epsilon^2} \log \frac{K}{\delta}\right)$ for any (ϵ, δ)-PAC algorithm for the ranking problem under the assumptions made by Binary-Search-Ranking on the preference relation $\mathbf{Q}$. For the best ranking problem in the PAC-setting, Falahatgar et al. (2017a) show that any algorithm needs $\Omega(K^2)$ comparisons under the strong stochastic transitivity property. To this end, they consider a preference relation for which they reduce the problem of finding a 1/4-ranking to a problem of finding a coin with bias 1 among $\frac{K(K-1)}{2} - 1$ other fair coins, showing that any algorithm requires at least a number of comparisons that scales quadratically with K. Finally, Falahatgar et al. (2018) verify the same lower bound by assuming moderate stochastic transitivity together with the stochastic triangle inequality. This in particular shows that the stochastic triangle inequality facilitates the learning problem.

3.1.14 Top-k identification via Quick Select

Ren et al. (2020) consider the (ϵ, δ)-PAC learning scenario for the top-k arms if the underlying preference relation has a total order over the arms, satisfies strong stochastic transitivity as well as the stochastic triangle inequality. Note that an ϵ-approximation of the top-k arms is any k-sized set of arms such that any arm within the set is ϵ-preferable to any other arm, which is not in the subset (cf. Section 2.2.5). Moreover, it is throughout assumed that k is at most $K/2$.

The suggested algorithm, called Tournament-k-Selection (T-k-S), is a tournament algorithm proceeding in rounds, each of which consists of dividing the currently selectable arms into subsets of sizes at most $2k$, and then using a subroutine called Epsilon-Quick-Select (EQS) on each subset. This is done to eliminate arms that are (with a specific confidence) not belonging to the (nearly) top-k arms of that subset. The entire process stops as soon as the number of noneliminated arms is k. The final output of T-k-S is then given by the remaining arms.

The subroutine EQS is inspired by the well-known QuickSelect algorithm (Hoare, 1961) to find the k largest items of an array. First, EQS chooses one arm randomly to be the pivot arm and then duels it with any other arm a specific number of times (similar as for each stage of Seq-Eliminate) leading to a partition of the set of arms into three subsets: One subset consisting of all arms one is confident enough that each of its elements are preferred resp. not preferred over the pivot arm, and one subset consisting of all arms one is not confident enough about the preference relation merged with the pivot arm itself. If the subset of surely preferred arms consists of more than k elements, the EQS algorithm is applied on the latter subset. Otherwise, if the union of surely preferred arms and unsurely preferred arms consists of more than k elements, then a k-sized subset is formed by merging the surely preferred arms and a randomly chosen subset (of the right size) of the unsurely preferred arms. Finally, if the latter union has strictly less than k elements, say k', then this union is returned together with the subset returned by EQS for finding the top-$(k-k')$ arms on the subset consisting of the surely not preferred arms.

By choosing the round-wise confidence levels and approximation errors in a suitable way, T-k-S is shown to be an (ϵ, δ)-PAC algorithm for finding the top-k arms with sample complexity $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{k}{\delta}\right)$. This sample complexity is optimal, as the authors show also a lower bound on the sample complexity of $\Omega\left(\frac{K}{\epsilon^2}\log\frac{k}{\delta}\right)$ for any (ϵ, δ)-PAC algorithm for finding the top-k arms under the assumptions made by T-k-S.

3.1.15 Approx-Prob

Falahatgar et al. (2018) consider the problem of approximating all pairwise probabilities to an accuracy of ϵ and present the Approx-Prob algorithm for this purpose with an optimal sample complexity of the form $\mathcal{O}\left(\frac{K\min(K,1/\epsilon)\log(K)}{\epsilon^2}\right)$ for $\delta \geq 1/K$. Approx-Prob takes a pre-sorted list of the items in the form of an $\epsilon/8$-ranking as an input—this requires solving a ranking problem first, for which the Binary-Search-Ranking algorithm as discussed in Section 3.1.13 can be used. Given this input, the algorithm reduces the number of pairwise comparisons by exploiting the assumptions of strong stochastic transitivity and the stochastic triangle inequality for the pairwise probabilities. The key idea is that, under these regularity assumptions, the pairwise probabilities should obey constraints of the form $\Delta_{i-1,j} \leq \Delta_{i,j} \leq \Delta_{i,j-1}$ for $i < j$. These constraints are used to correct empirical estimates $\hat{\Delta}_{i,j}$, which are obtained as relative winning frequencies from repeated pairwise comparisons of arms a_i and a_j, or even to avoid such an estimation altogether. For example, if $\hat{\Delta}_{i-1,j} = \hat{\Delta}_{i,j-1}$, then by virtue of the above constraint, inequalities turn into equalities $\Delta_{i-1,j} = \Delta_{i,j} = \Delta_{i,j-1}$, and $\hat{\Delta}_{i,j}$ is simply set to $\hat{\Delta}_{i-1,j}$ instead of being estimated. Such equalities are fostered by providing estimates on a grid, i.e., estimates of pairwise probabilities are always rounded off to the closest multiple of ϵ. Moreover, to take the greatest advantage of the triplet-constraints, the items are compared in a specific order: a_i is compared to a_j in an outer loop for $i = 1, \ldots, K-1$ and an inner loop for $j = i+1, \ldots, K$.

3.1.16 Robust Query Selection

Jamieson and Nowak (2011) assume that there is a total order over the arms and each arm can be embedded into $\mathbb{R}^d$ by means of suitable location points. Further, they assume that there exists some (reference) point in $\mathbb{R}^d$, such that the Euclidean distance of this point to these locations is coherent with the underlying total order in the sense that the closer an arm's location point is to the reference point, the higher its ranking in the underlying total order. While the locations are assumed to be known, the reference point is unknown. By first assuming that the outcome of a duel between two arms is deterministic, i.e., $\mathbf{Q} \in \{0,1\}^{K \times K}$, the authors introduce the notion of ambiguity of a duel: If it is not possible to infer from the past duels of other pairs of arms which arm will be preferred over the other for a specific pair of arms, then the latter pair is called *ambiguous*, otherwise it is called *unambiguous*.

In order to characterize the property of ambiguity in the presence of a (latent) reference point and the embedding of the arms, they relate the problem of identifying the underlying ranking of the arms to the problem of determining the label of $(d+1)$-dimensional points via linear separators (in an active way). With this, they introduce the Query Selection algorithm, which essentially samples a pair of arms uniformly at random from the set of pairs not considered so far, checks whether the pair is unambiguous in order to skip superfluous duels, and carries out the duel in case the pair is ambiguous. Once all pairs have been considered, the algorithm terminates and provably returns the correct ranking.

For the scenario in which the outcome of duels between two arms is not necessarily deterministic, the latter algorithm is modified to the Robust Query Selection algorithm. This algorithm essentially corresponds to Query Selection. However, due to the noisy outcomes, it conducts a pre-specified number of duels for an ambiguous pair of arms in order to decide which arm is preferred over the other, namely the arm which has won the majority of the noisy duels. It is shown that if the number of duels carried out per ambiguous pair is set to $\log(2K\log(K)/\delta)/2h^2$, where $\delta \in (0,1)$ and $h \in (0,1/2)$ is such that $\min_{1 \le i < j \le K} |\Delta_{i,j}| \ge h$, then the Robust Query Selection algorithm returns the true underlying ranking and has a sample complexity of order $\mathcal{O}\left(\frac{d}{h^2}\log^2\left(\frac{K}{\delta}\right)\right)$.

3.1.17 Verification-based Solution

Karnin (2016) considers the problem of finding the best arm in structured MAB problems, which refers to a general bandit framework covering a variety of different bandit problems such as the classical MAB problem, linear bandits, combinatorial bandits and other bandit problems (see Lattimore and Szepesvári (2020) for an overview). To this end, a general algorithmic framework is introduced, consisting of two subroutines which need to be specified for each underlying bandit problem. The first subroutine, referred to as FindBestArm, needs to be designed such that it identifies the best arm with a certain high probability, while using as few samples within the underlying bandit problem as possible. The second subroutine, referred to as VerifyBestArm, serves the purpose of verifying the optimality of the arm suggested by the first subroutine as the best arm with a certain degree of confidence. In particular, this subroutine can either confirm or refuse the optimality of the suggested arm and can also receive additional information about the underlying bandit problem from the first subroutine. Both subroutines are run one after the other during iterative stages, where in each stage the confidence level for the second subroutine is geometrically decreased, while the error probability of the first subroutine is throughout a specific constant. The iteration process transitions into the next stage only if the second subroutine has refused the optimality of the suggested arm, and the entire process stops as soon as the second subroutine has confirmed the optimality of the suggested arm. The rationale behind this procedure is that it seems to be easier to verify or refute the optimality of a candidate arm than to explicitly search for the best arm.

In concrete terms for the setting of dueling bandits under the Condorcet assumption, a suggestion for both subroutines is made. The suggestion for the first subroutine initializes an active set consisting of all ordered pairs of arms, and duels all active pairs, say (a_i, a_j), until one of three specific conditions on the lower resp. upper confidence estimates of the underlying pairwise preference probabilities, i.e., $\widehat{q}_{i,j}^t - c_{i,j}^t$ resp. $\widehat{q}_{i,j}^t + c_{i,j}^t$, holds, whereupon the pair is removed from the active set. Once the active set is empty, the subroutine stops and returns as its candidate for the Condorcet winner the arm that has a lower confidence estimate greater than $1/2$ against any other arm. The three conditions are designed such that this condition is fulfilled for exactly one arm, which is likely to be the actual Condorcet winner. Here, $c_{i,j}^t$ is a confidence interval based on Hoeffding's inequality for maintaining the predefined error probability of the first subroutine. The second subroutine proceeds from the information available after the first subroutine, i.e., the candidate for the Condorcet winner as well as for each allegedly non-Condorcet winner arm its toughest competitor (similarly as for RCS), and the verification task consists of verifying that each non-Condorcet winner arm is indeed inferior to its toughest competitor based on confidence intervals maintaining the confidence level for the second subroutine. It is shown that these two subroutines used in the general algorithmic framework lead to a sample complexity of order

$$\mathcal{O}\left(\sum\nolimits_{i\neq i^*} \min_{j: q_{i,j}<1/2} \frac{\log\left(K/(\delta \Delta_{i,j}^2)\right)}{\Delta_{i,j}^2}\right) + \tilde{\mathcal{O}}\left(\sum\nolimits_{i\neq i^*}\left(\Delta_{i^*,i}^{-2} + \sum\nolimits_{j\neq i}\Delta_{i,j}^{-2}\right)\right), \tag{4}$$

where $\tilde{\mathcal{O}}$ is hiding log-factors. By considering the RMED algorithm of Komiyama et al. (2015) as an algorithm for sample complexity minimization and transforming its expected regret upper bound to a sample complexity bound, the method suggested by Karnin (2016) achieves an improvement over the latter by a multiplicative factor K^ϵ, provided δ is sufficiently large.

3.1.18 Parallel Selection and Partition

Under the assumption of a total order over the arms, the task of finding the kth best arm or a partition of $\mathcal{A}$ into the set of best k arms and its complement is considered by Braverman et al. (2016). The authors are not only interested in the sample complexity of algorithms for the latter task, but also in their round complexity, which can be interpreted as the degree of an algorithm's adaptivity. To be more precise, an algorithm making all decisions on the order of the duels entirely ahead of time has the smallest possible round complexity (non-adaptive), while an algorithm deciding in each time step which duel to conduct has the largest possible round complexity (fully adaptive).

The authors provide algorithms that correctly solve the above tasks with high probability, having a low round complexity, while the sample complexity is of order

$$\mathcal{O}\left(\frac{K}{\min_{1\leq i<j\leq K}\Delta_{i,j}^2}\log(K)\right),$$

assuming the knowledge of $\min_{1 \leq i < j \leq K} \Delta_{i,j}^2 > 0$. Further, it is shown that any algorithm, which can correctly return the kth best arm with high probability, has a sample complexity of the same order as above, revealing the optimality of the suggested algorithmic solutions.

Just as for the Robust Query Selection algorithm in Section 3.1.16, the authors first consider the case of deterministic outcomes of duels in order to design suitable algorithmic solutions and transfer them to the probabilistic scenario by repeating a duel between a specific pair of arms a certain number of times. The major algorithmic procedure underlying the deterministic case first chooses a sufficiently large subset of $\mathcal{A}$ in a random manner, say $\mathcal{A}^S$, and then reduces $\mathcal{A}^S$ successively by choosing random anchor arms in $\mathcal{A}^S$, comparing these with all arms in $\mathcal{A}^S$ and keeping all arms that are in some sort of interquartile range (with respect to the number of duels won) of the anchor arms. The rationale behind the latter iterative process is that the subset $\mathcal{A}^S$ will reduce quickly to the kth best arm (or a small subset containing it) of the initially first chosen subset, which in turn is likely to be close[7] to the kth best arm in $\mathcal{A}$. Hence, one (randomly chosen) arm within the reduced subset can be used as a kind of pivot in order to partition $\mathcal{A}$ into the set of best k arms and its complement by dueling the former with all arms in $\mathcal{A}$: All arms preferred over the pivot are in the top set, while the inferior arms are in its complement.

By running the major procedure twice with suitable choices for the subset sizes as well as number of iterations, and then combining and filtering the results of both procedures, the authors show that the true partition (and also the kth best arm) can be identified with high probability.

3.1.19 Single Elimination Tournament

Assuming the existence of a total order over the arms, Mohajer et al. (2017) study the top-k identification problem as also considered by Braverman et al. (2016), and additionally the top-k ranking problem, both with the goal to minimize the exact sample complexity while maintaining a predefined confidence δ. To this end, they first present the Select algorithm for identifying the best arm, which can be seen as a customized single-elimination tournament consisting of multiple layers, where in each layer, pairs of arms are randomly built first, and on the basis of pairwise comparisons, one arm is retained and the other one is eliminated. This process is repeated until only one arm is left, which is then the suggestion for the best arm. Note that the Select algorithm is conceptually similar to Knockout (cf. Section 3.1.10). The authors subsequently show that the algorithm finds the best arm with probability at least $1 - \delta$ and has a sample complexity of order $\mathcal{O}(\frac{K \log(1/\delta)}{\Delta_{1,2}^2})$.

Select is then generalized to the Top algorithm, which works for both top-k ranking and identification, by first splitting the arms into k sub-groups, then identifying the best arm in each sub-group using Select, and finally forming a short list that includes all winners from the sub-groups. For this list, they build a heap data structure, from which the top-k arms are extracted one after another, while whenever a best arm is extracted from its list, the second best arm from that list is identified and reinserted into the short list. Unfortunately, it is not shown by the authors how the sample complexity of Top scales in terms of δ,. We conjecture it is of the order $\mathcal{O}\left(\frac{(K+k\log k)\ \log(1/\delta)\ \max\{\log k, \log\log K\}}{\log\log K\ \Delta_k}\right)$, where

7. Closeness is here to be understood in terms of their ranks with respect to the ground truth ranking.

$\Delta_k = \min_{i \in [k]} \min_{j: j \geq i} \Delta_{i,j}^2$ in the case of top-k ranking and $\Delta_k = \Delta_{(k),(k+1)}^2$ in the case of top-k identification, with $\Delta_{(k),(k+1)}$ denoting the calibrated pairwise preference probability of the arm with rank k and the arm with rank $k+1$ according to the total order of the arms. In Table 3, we report the sample complexity as stated by the authors.

Similarly as in the (ϵ, δ)-PAC setting, the leading factor of the sample complexity of Top differs from the one of SELECT by a logarithmic factor.

3.1.20 Sequential-Elimination-Exact-Selection

Having the same goal as Mohajer et al. (2017) but imposing stricter assumptions on $\mathbf{Q}$ (strong stochastic transitivity and stochastic triangle inequality), the authors of the T-k-S algorithm also propose the Sequential-Elimination-Exact-Best-Selection (SEEBS) algorithm for finding the best arm and the Sequential-Elimination-Exact-k-Selection (SEEKS) for identifying the top-k arms.

The SEEBS algorithm is a challenge algorithm proceeding in rounds. In each round the T-k-S algorithm is used to obtain a nearly best arm by setting k to 1, which is used as the current "champion". The latter arm is then used in a similar manner as the pivot arm in the QuickSelect inspired subroutine EQS of T-k-S to partition the set of arms into three subsets (surely preferred, surely not preferred and undecided arms). All arms ("challengers") assigned to the set of surely not preferred arms over the champion are eliminated. Adapting the confidence for the latter assignment as well as for the choice of the round-wise champion in an appropriate way over the rounds will ensure that eventually only one single arm is left and SEEBS maintains the overall confidence level δ. Moreover, SEEBS reveals an expected sample complexity of $\mathcal{O}\left(\sum_{i \in [K]} \frac{\log 1/\delta + \log\log 1/\Delta_{i,(k)}}{\Delta_{i,(k)}^2}\right)$, which is shown to be nearly optimal by providing a lower bound of $\Omega\left(\sum_{i \in [K]} \frac{\log 1/\delta}{\Delta_{i,(k)}^2} + \log\log 1/\Delta_{i,(k)}\right)$ for any algorithm able to identify the top-k arms with probability at least $1-\delta$ under the above assumptions on $\mathbf{Q}$. Here, the gaps $\Delta_{i,(k)}$ are defined for an arm a_i by the calibrated pairwise preference probability of a_i and the $(k+1)$st best arm if a_i belongs to the top-k arms, and otherwise by the calibrated pairwise preference probability of a_i and the kth best arm.

SEEKS is proceeding in rounds, just like SEEBS, but uses two versions of T-k-S to successively build a pool of candidate arms for the top-k set: The first variant is used to extract at the beginning of each round a smaller subset of all currently selectable arms with the size equal to the number of remaining arms for the final top-k set, while the second variant is used to determine the nearly worst arm of the latter subset[8]. The nearly worst arm is then used in a similar manner as the round-wise "champion" of the SEEBS algorithm to partition the initially extracted subset of arms into three subsets (surely preferred, surely not preferred and undecided arms) and include all surely preferred arms into the pool of candidates, and eliminate all surely not preferred *and* the surely preferred arms from the list of selectable arms for the next round. The procedure stops as soon as either the pool of candidates consists of at least k arms, or the cardinality of the candidate pool and still selectable arms is at most k. In the first termination event, a randomly chosen

8. Here, nearly worst is once again to be understood in terms of the rank in the underlying total order of the arms.

k-sized subset of the candidate pool is the final top-k set, while in the second termination event the candidate pool augmented by a random subset of the selectable arms in order to form a k-sized subset is used for the final top-k set. Once again, adapting the round-wise confidence levels in an appropriate way over the rounds leads to a sample complexity of $\mathcal{O}\left(\sum_{i\in[K]} \frac{\log K/\delta+\log\log 1/\Delta_{i,(k)}}{\Delta^2_{i,(k)}}\right)$ for SEEKS.

3.1.21 Iterative-Insertion-Ranking

Assuming that a total order over the arms exists and shall be found, Ren et al. (2019) leverage the idea of the BinarySearch algorithm (Feige et al., 1994) in order to deal with the case of an unknown lower bound for the calibrated pairwise preference probabilities $|\Delta_{i,j}|$. To this end, they introduce the Iterative-Insertion-Ranking (IIR) algorithm, which essentially is a binary insertion sort to rank the arms, where the underlying tree structure is a *preference interval tree* (PIT) as also used in BinarySearch.

A PIT is a binary tree where each node n maintains a triple of elements $n[1], n[2], n[3]$, each of which is an element of $\mathcal{A} \cup \{-\infty, \varnothing, +\infty\}$, where $-\infty, \varnothing, +\infty$ are to be understood as symbols representing artificial arms. For each inner node n, it holds that $n[2] \neq \varnothing$, while for all leaf nodes l, the second element $l[2]$ is always set to $\varnothing$. Further, each inner node n has a pointer to its children $n.left$ and $n.right$, respectively, and it holds that $n[1] = n.left[1]$, $n[3] = n.right[3]$ as well as $n[2] = n.left[3] = n.right[1]$. The root node n_0 of the PIT is such that $n_0[1] = -\infty$ and $n_0[3] = +\infty$. Given these conditions, the idea is that the sequence of leaf nodes from left to right represent the underlying total order of the arms: If $l_1, \ldots, l_{K+1}$ are the leaf nodes from left to right, then the total order of the arms is given by $l_2[1] \prec l_3[1] \prec \ldots \prec l_{K+1}[1]$ or equivalently $l_1[3] \prec l_2[3] \prec \ldots \prec l_K[3]$.

The IIR algorithm successively builds a PIT by inserting each arm one after the other, once being confident enough about its right position, by creating leaf nodes and updating the inner nodes accordingly. To this end, the algorithm is repeatedly trying to insert an arm a_i into the PIT by moving through the PIT starting from the root and successively dueling a_i with the (non-artificial) arms of the current node in order to determine the direction of the next move, or to insert the arm into the PIT — both based on certain confidence levels. Once a leaf node is created for the current arm, the insertion mechanism is started for a not yet included arm until all arms are included in the PIT. The used confidence levels are based on a guess on $\min_{j\neq i} \Delta_{i,j}$, and it is shown that the insertion of an arm is at the right position of the PIT with high probability once the guess is appropriate, while the guess is updated in case the arm cannot be inserted with certainty into the PIT. In the latter case, the insertion mechanism is started from scratch with the same arm.

A sample complexity of $\mathcal{O}\left(\sum_{i\in[K]} \frac{(\log\log(\min_{j\neq i}\Delta_{i,j}^{-1})+\log(K/\delta))}{\min_{j\neq i}\Delta^2_{i,j}}\right)$ is derived for IIR in order to guarantee that the true underlying ranking is represented by the final PIT with probability at least $1-\delta$. This sample complexity is nearly optimal if strong stochastic transitivity holds, as the authors derive a lower bound on the sample complexity of any learner to return the true underlying ranking maintaining the confidence level δ of

$$\Omega\left(\sum_{i\in[K]} \frac{(\log\log(\tilde{\Delta}_i^{-1}) + \log(1/\delta))}{\min_{j\neq i}\tilde{\Delta}_i^2} + \inf\left\{\sum_{i\in[K]} \tilde{\Delta}_i^{-2}\log(1/p_i) \,\Big|\, \sum_{i\in[K]} p_i \leq 1\right\}\right),$$

where $\tilde{\Delta}_i$ denotes the smaller of the two calibrated pairwise preference probabilities of an arm a_i to its (two) adjoining arms according to the underlying true ranking. In particular, $\tilde{\Delta}_i$ and $\min_{j \neq i} \Delta_{i,j}$ are in general not the same, but coincide if $\mathbf{Q}$ satisfies strong stochastic transitivity.

3.2 Regularity Through Latent Utility Functions

The representation of preferences in terms of utility functions has a long history in decision theory (Fishburn, 1970). The idea is that the absolute preference for each arm can be reflected by a real-valued utility degree.

This idea applied to the dueling bandits setting is also known as *utility-based dueling bandits,* where one assumes the existence of a latent utility function $u : \mathcal{A} \rightarrow \mathbb{R}$ with $u(a_i)$ representing the utility of an arm $a_i \in \mathcal{A}$. Further, the probability of the outcome of a pairwise comparison between two arms a_i, a_j is determined by the difference of their utilities. However, as the difference is not necessarily a value inside the unit interval, one makes use of a link function $\sigma : \mathbb{R} \rightarrow [0,1]$ in order to map these differences of utilities to actual probabilities. Formally,

$$\mathbf{P}(a_i \succ a_j) = \sigma\big(u(a_i) - u(a_j)\big), \tag{5}$$

so that $\Delta_{i,j} = \sigma\big(u(a_i) - u(a_j)\big) - 1/2$ in this setting. The minimal assumptions on the link function are as follows:

1. strict monotonicity on $\sigma^{-1}(0,1)$, so that an arm with a higher utility than another arm will also have a higher probability to be chosen than the latter;
2. $\sigma(0) = 1/2$, which implies that two arms having the same utility have also the same chance to beat the other one, respectively.

In principle, any cumulative distribution function of a symmetric continuous random variable is in line with these conditions. The two most common link functions, which are both satisfying the conditions, are the *logistic link function* $\sigma(x) = 1/(1 + \exp(-x))$ and the *linear link function* $\sigma(x) = \max\{0, \min\{1, 1/2 \cdot (1 + x)\}\}$.

Under the above assumptions on the link function and the additional assumption that the utility function is injective, the low noise model assumption holds for $\mathbf{Q}$. Indeed, injectivity of u implies for distinct arms $a_i, a_j \in \mathcal{A}$ that

$$\Delta_{i,j} = \sigma\big(u(a_i) - u(a_j)\big) - 1/2 \neq 0\,.$$

Without loss of generality, let $u(a_i) > u(a_j)$. Then, by the strict monotonicity of σ, we obtain

$$\Delta_{i,k} = \sigma\big(u(a_i) - u(a_k)\big) - 1/2 > \Delta_{j,k} = \sigma\big(u(a_j) - u(a_k)\big) - 1/2$$

for all $k \in [K] \setminus \{i, j\}$. Thus, summing over all k shows that the low noise model assumption is fulfilled. Note that in case u is not injective, the implication does not hold in general and also a total order over the arms does not exist. However, one can still infer that SST is satisfied for the corresponding preference relation $\mathbf{Q}$.

Another appealing property of the approach based on utility functions is the possibility to model the dueling bandits problem for the case of infinitely many arms. Let $\mathcal{S}$ be some space of arms, which is not necessarily finite[9]. Then, one assumes the existence of $u : \mathcal{S} \to \mathbb{R}$ with $u(a)$ representing the utility of an arm $a \in \mathcal{S}$. With this, the pairwise comparison probabilities are modeled as in (5) by means of some suitable link function σ. Assume u to be injective and σ to satisfy the requirements above. Then, similar to the case of finitely many arms, one can infer that the low noise model assumption for the infinite arm scenario holds, while in case of a non-injective u, SST still holds [10].

In summary, the assumption of a latent utility function for the arms is a stronger assumption than the regularity properties in Section 3.1, provided the utility function is injective. However, due to the increased structural complexity of a space of infinitely many arms, the utility assumption seems to be necessary, as it provides a surrogate for the quantitative rewards making the learning possible based on merely qualitative feedback.

The regret definition in the utility-based dueling bandits scenario is similar to the one in (2) and given by

$$R^T = \sum_{t=1}^{T} \Delta_{a_*,a_t} + \Delta_{a_*,a_t'} = \sum_{t=1}^{T} \sigma(u(a_*) - u(a_t)) + \sigma(u(a_*) - u(a_t')) - 1 \ , \tag{6}$$

where (a_t, a_t') is the pair of arms chosen in time step t. Here, however, two variants are considered for the reference arm a_* :

(i) a_* is the best one known only in hindsight, which is inspired by the classical regret definition in the realm of online learning with full information.

(ii) a_* is an arm with highest utility, i.e., $a_* \in \begin{cases} \mathrm{argmax}_{a\in\mathcal{A}}\, u(a), & \text{finite arm case,} \\ \mathrm{argmax}_{a\in\mathcal{S}}\, u(a), & \text{infinite arm case.} \end{cases}$

The problem of finding the best arm in hindsight can be viewed as a noisy online optimization task (Finck et al., 2011), where the underlying search space is $\mathcal{S}$, and the function values cannot be observed directly; instead, only noisy pairwise comparisons of function values (utilities) are available. In this framework, it is hard to have a reasonable estimate for the gradient, so that classical online optimization algorithms are not directly applicable.

In the following, we investigate existing methods for PB-MAB problems that are posing a latent utility structure on the available arms. For reasons of clarity, we list all these approaches in Table 4 in the spirit of the previous section.

3.2.1 Dueling Bandits Gradient Descent

The dueling bandits setting has originally been introduced by Yue and Joachims (2009). The authors consider the case of a possibly infinite number of arms, which are all elements of

9. This space corresponds to our set of arms $\mathcal{A}$. However, as we assume $\mathcal{A}$ to be finite, we use another notation here.

10. Here, a suitable substitute for the sum occurring in the low noise model assumption is necessary. One possible approach is available in the case, where $(\mathcal{S}, \mathbb{S}, \mu)$ is a measure space. Then, an extension of the low noise assumption can be defined as follows: For all $a, \tilde{a} \in \mathcal{S}$ with $a \neq \tilde{a}$, it holds that $\Delta_{a,\tilde{a}} \neq 0$, and if $\Delta_{a,\tilde{a}} > 0$, then

$$\int_{\mathcal{S}} \Delta_{a,a'} \mathrm{d}\mu(a') > \int_{\mathcal{S}} \Delta_{\tilde{a},a'} \mathrm{d}\mu(a').$$

Algorithm	Algorithm class(es)	Assumptions	Target(s) and goal(s) of learner	Theoretical guarantee(s)
Dueling Bandit Gradient Descent (Section 3.2.1)	Online optimization based	Strictly concave, Lipschitz-continuous utility function and rotation-symmetric, second order Lipschitz-continuous link function	Expected regret minimization with best arm in hindsight	$\mathcal{O}\left(T^{3/4}\sqrt{d}\ \right)$
Noisy Comparison-based Stochastic Mirror Descent (Section 3.2.2)	Online optimization based	Smoothness and concavity assumptions for the utility functions; differentiability assumption & shape-constraints on link function	Expected regret minimization with best arm in hindsight	$\mathcal{O}\left(d\sqrt{T}\,\log T\right)$
Doubler (Section 3.2.3)	Reduction-based	Linear link function	Expected regret minimization for best arm	Finite arms: $\mathcal{O}\left(\frac{K\log^2 T}{\min_{j\neq i^*}\Delta_{i^*,j}}\right)$ Infinite arms: $\mathcal{O}\left(\frac{d^2\log^4 T}{\min_{a\neq a_*}\Delta_{a_*,a}}\right)$ resp. $\mathcal{O}\left(\sqrt{d\,T\,\log^3(T)}\right)$
MultiSBM (Section 3.2.3)	Reduction-based	Linear link function	Expected regret minimization for best arm	Finite arms: $\mathcal{O}\left(\sum_{i\neq i^*}\frac{\log T+K\log(K)}{\Delta_{i^*,j}}\right)$
Winner stays (Section 3.2.4)	Tournament	Linear link function	Expected weak and strong regret minimization for best arm	Weak regret: $\mathcal{O}\left(\frac{K\log K}{\min_{i,j}\Delta_{i,j}^5}\right)$ Strong regret: $\mathcal{O}\left(K\log K+K\log T\right)$
Dueling Bandits Temporary Elimination Algorithm (Section 3.2.5)	Generic tournament	Linear link function	Expected regret minimization for best arm	$\mathcal{O}\left(\sum_{i\neq i^*}\frac{\log T}{\Delta_{i^*,j}}\right)+\mathcal{O}\left(K\right)$
Round-Efficient Dueling Bandits (Section 3.2.6)	Tournament	Linear link function	1. Sample and round complexity minimization for best arm 2. (ϵ,δ)-PAC variant of 1.	1. Round complexity: $\mathcal{O}\left(\frac{\log\left(\frac{K}{\delta\,\min_{j\neq i^*}\Delta_{i^*,j}}\right)}{\min_{j\neq i^*}\Delta_{i^*,j}^2}\right)$ Sample complexity: $\mathcal{O}\left(\sum_{j\neq i^*}\frac{\log\left(\frac{K}{\delta\,\Delta_{i^*,j}}\right)}{\Delta_{i^*,j}^2}\right)$ 2. Round complexity: $\mathcal{O}\left(\frac{\log\left(\frac{K}{\delta\,\epsilon}\right)}{\epsilon^2}\right)$ Sample complexity: See (7)
Multisort (Section 3.2.7)	Noisy-sorting	Logistic link function and well-separated utilities	Sample complexity minimization for total ranking	$\mathcal{O}\left(K\log^6 K\right)$

Table 4: Utility-based approaches for the dueling bandits problem. In the case of infinitely many arms, the dimension of the space of arms is denoted by d, while i^* is the index representing the best arm in case of finite arms and a^* for infinite arms.

a compact, convex subset of $\mathbb{R}^d$ containing the origin. Further, they assume the existence of a strictly concave, Lipschitz-continuous utility function u of the arms, as well as a rotation-symmetric, second-order Lipschitz-continuous link function σ. The goal of the learner is to minimize the (expected) regret defined as in (6), where the reference arm is the best arm in hindsight.

As mentioned above, this problem can be seen as an online noisy optimization task, where, however, a reasonable estimate for the gradient is lacking. Therefore, the authors opt for applying an online convex optimization method (Flaxman et al., 2005), which does not require the gradient to be calculated explicitly, and instead optimizes the parameter by estimating an unbiased gradient approximation. The suggested algorithm, called Dueling Bandit Gradient Descent (DBGD), is an iterative search method traversing the space $\mathcal{S}$ over the time horizon $\{1, 2, \ldots, T\}$. For a time step t, let $a_t \in \mathcal{S}$ be the current point. To determine the dueling arm for a_t, a random direction u_t is drawn uniformly from the unit ball and a'_t is set to be $\Pi_{\mathcal{S}}(a_t + \delta u_t)$, where $\Pi_{\mathcal{S}}(\cdot)$ denotes the projection into $\mathcal{S}$, and δ is some step parameter steering the degree of exploration. This point a'_t is compared with the current point a_t, with two possible consequences for the update of the algorithm: if a_t wins the comparison, the subsequent point a_{t+1} is set to be a_t, while in the case of a defeat, the subsequent point is $\Pi_{\mathcal{S}}(a_t + \gamma u_t)$ with $\gamma > 0$. Here, γ is an exploitation parameter of the method, as it determines the step size with which an update is taken into the winner direction.

Assuming that the space of arms $\mathcal{S}$ is a d-dimensional ball of radius R, the expected regret of the proposed method is shown to be bounded by $\mathbf{E}[R^T] \leq 2T^{3/4}\sqrt{10RdL}$ for an appropriate choice of δ and γ, where L is the product of the Lipschitz constants of the link function σ and the latent utility function u.

3.2.2 Noisy Comparison-based Stochastic Mirror Descent

Kumagai (2017) studies the utility-based dueling bandits problem imposing convexity and smoothness assumptions for the utility function, which are stronger than those by Yue and Joachims (2009), and which guarantee the existence of a unique minimizer of the utility function. Other assumptions concern the link function, which are weaker than those by Ailon et al. (2014b) (cf. Section 3.2.3) and satisfied by common functions, including the logistic function used in the experiments by Yue and Joachims (2009), the linear function used by Ailon et al. (2014b), and the Gaussian distribution function.

Motivated by the fact that Yue and Joachims (2009) use a stochastic gradient descent algorithm for the utility-based dueling bandits problem, the authors propose to use a stochastic mirror descent algorithm, which is known to ensure near optimality in convex optimization problems. To this end, the expected regret minimization problem of a utility-based dueling bandits problem is reduced to a locally-convex optimization problem, for which the proposed algorithm, called Noisy Comparison-based Stochastic Mirror Descent (NC-SMD) can be analyzed in the spirit of bandit convex optimization problems (Hazan, 2016; Kumagai, 2018). With this, it is shown that NC-SMD achieves a regret bound of $\mathcal{O}\left(d\sqrt{T\log T}\right)$ in expectation if the set of arms $\mathcal{S}$ is a compact convex subset of $\mathbb{R}^d$ with non-empty interior and the learning rate of NC-SMD is suitably tuned (i.e., depending on the time horizon

T). The latter regret bound is optimal except for a logarithmic factor, as they derive a lower bound of order $\Omega(d\sqrt{T})$ by relating their problem to a convex optimization problem.

3.2.3 Reduction to Value-based MAB

Ailon et al. (2014b) propose various methodologies to reduce the utility-based PB-MAB problem to the standard value-based MAB problem for the case of finitely many arms (denoted by $\mathcal{A}$) as well as for infinitely many arms (denoted by $\mathcal{S}$). In their setup, the utility of an arm is assumed to be in $[0, 1]$. Formally, $u : \mathcal{S} \to [0, 1]$, and the link function is the linear link function $\sigma(x) = \max\{0, \min\{1, 1/2 \cdot (1 + x)\}\}$. Therefore, the probability of an arm $a \in \mathcal{S}$ beating another arm $a' \in \mathcal{S}$ is

$$\mathbf{P}(a \succ a') = \frac{1 + u(a) - u(a')}{2} \quad ,$$

which is again in $[0, 1]$. The regret considered is the one defined in (6), where the reference arm a_* is the globally best arm with maximal utility.

Ailon et al. (2014b) propose two reduction techniques, Doubler and MultiSBM, for a finite and an infinite set of arms. In both techniques, value-based MAB algorithms such as UCB (Auer et al., 2002a) are used as a black box for driving the search in the space of arms. For a finite number of arms, value-based bandit instances are assigned to each arm, and these bandit algorithms are run in parallel. More specifically, assume that an arm $i(t)$ is selected in iteration t (to be explained in more detail shortly). Then, the bandit instance that belongs to arm $i(t)$ suggests another arm $j(t)$. These two arms are then compared in iteration t, and the reward, which is 0 or 1, is assigned to the bandit algorithm that belongs to $i(t)$. In iteration $t+1$, the arm $j(t)$ suggested by the bandit algorithm is compared, that is, $i(t+1) = j(t)$. What is nice about this reduction technique is that, under some mild conditions on the performance of the bandit algorithm, the preference-based expected regret defined in (2) is asymptotically identical to the one achieved by the value-based algorithm for the standard value-based MAB task.

For infinitely many arms, the reduction technique can be viewed as a two player game. A run is divided into epochs: the ℓth epoch starts in round $t = 2^{\ell}$ and ends in round $t = 2^{\ell+1} - 1$, and in each epoch the players start a new game. During the ℓth epoch, the second player acts adaptively according to a strategy provided by the value-based bandit instance, which is able to handle infinitely many arms, such as the ConfidenceBall algorithm by Dani et al. (2008). The first player obeys some stochastic strategy, which is based on the strategy of the second player from the previous epoch. That is, the first player always draws a random arm from the multi-set of arms that contains the arms selected by the second player in the previous epoch. This reduction technique incurs an extra $\log T$ factor to the expected regret of the value-based bandit instance.

3.2.4 Winner Stays

The WS-W resp. WS-S algorithm (Chen and Frazier, 2017), which was already discussed in Section 3.1.8, has also been analyzed by the authors for the case of a utility-based dueling bandits scenario with a linear link function and finitely many arms. In particular, the weak regret resp. strong regret version of (6) are considered (cf. discussion following (2)) and a bound of order $\mathcal{O}\left(\frac{K \log K}{\min_{i,j} \Delta_{i,j}^5}\right)$ is derived for WS-W for minimizing weak cumulative regret, while for minimizing strong cumulative regret a bound of order $\mathcal{O}\left(K \log K + K \log T\right)$ in expectation is inferred for WS-S.

3.2.5 Dueling Bandits Temporary Elimination Algorithm

Zimmert and Seldin (2018) introduce the general identifiability assumption (cf. Section 3.1) and show that any link function as used by Yue and Joachims (2009) fulfills this assumption. Like Ailon et al. (2014b), they study the dueling bandits problem for a linear link function with the goal of expected regret minimization. For this purpose, they consider the sub-optimality gaps $\tilde{\Delta}_a := \min_{a' \neq a^*} \Delta_{a^*,a'} - \Delta_{a,a'}$, where a^* is an identifiable arm emerging in the general identifiability assumption, which allows them to rewrite the expected regret in (6) as

$$\mathbf{E}[R_A^T] = \sum_{t=1}^{T} \mathbf{E}[\delta(a_*, a_t) + \delta(a_*, a_t')] = \sum_{a \neq a^*}^{T} \mathbf{E}[N_T(a)]\, \tilde{\Delta}_a \ ,$$

where $N_T(a)$ is the total number of plays of an arm a.

Based on this relationship between the regret and the sub-optimality gaps, they suggest the Dueling Bandits Temporary Elimination Algorithm (DBTEA), which is a phase-based anytime algorithm maintaining a set of active arms consisting of arms having a non-positive lower confidence bound on the sub-optimality gap. For these arms, one is not yet certain enough that the sub-optimality gap is indeed positive (it is only zero for optimal arms). Each phase consists of dueling arms from two random orders of the active set, such that each arm in the active set duels twice in a phase.

Exploiting the relationship of the regret per time and the sum of the sub-optimality gaps is at the heart of their approach, so that a different proof technique is required to obtain confidence bounds for the latter sum. For this purpose, they provide a novel anytime concentration inequality for the sum of sub-Gaussian random variables.

The authors' theoretical analysis of the DBTEA leads to an upper bound on its expected regret of $\mathcal{O}(K) + \mathcal{O}(\sum_{i \neq i^*} \frac{\log(T)}{\tilde{\Delta}_i})$. In numerical experiments, they illustrate that the additive term appearing in some of the theoretical upper bounds of state of the art algorithms (cf. Table 1) might dominate the regret for problem instances with a large number of arms and moderate time horizons, so that the DBTEA is superior to some state of the art algorithms in these cases.

The setting in Zimmert and Seldin (2018) is in fact more general, as they introduce the *factorized bandit problem*, in which each action consists of a Cartesian product of elementary actions. This is a generalization of the so-called rank-1 bandit problem (Katariya et al., 2017), and allows to model the dueling bandits setting by considering $\mathcal{A} \times \mathcal{A}$ as the action space, i.e., the action is to "generate" a duel between two arms.

3.2.6 Round-Efficient Dueling Bandits

Inspired by real world scenarios such as sport tournaments, where participants simultaneously compete against each other at a time, Lin and Lu (2018) consider the dueling bandits scenario where the learner maintains in each iteration a set of active arms, builds a set of disjoint pairs of these arms and obtains all (noisy) pairwise preferences of these pairs. As a matter of fact, this problem scenario can be seen as a special case of the multi-dueling bandit problem, which will be discussed in Section 6, since the learner obtains preference feedback in the form of a sequence of pairwise preferences, or in other words a partial preference feedback (cf. Section 6.2.4). However, as the authors assume a latent utility for each arm and a linear link function underlying the probability in (5), their problem scenario fits quite well into the scope of this section.

The goal is to find the best arm while keeping the total number of rounds as well as the total number of pairwise comparisons as small as possible. Here, the number of rounds is simply the number of iterations, i.e., how many times the learner carries out its action in the form of a pairwise comparison of the pairs of active arms. Due to the underlying latent utilities, the best arm is the one with the highest utility. The authors derive unbiased estimates and corresponding confidence intervals for the latent utilities of the active arms, and based on these design an algorithm which successively eliminates arms in the active set as soon as their estimated utility is below the lower confidence utility of the empirically best arm. At the beginning, apparently all arms are active and the algorithm simply partitions all available arms randomly into disjoint pairs. If the size of active arms is an odd number, i.e., there is one arm without a dueling partner, this arm is not compared with another arm until one of the arms is eliminated, whereupon the former builds a pair with the arm that competed with the eliminated one.

It is shown that this algorithm returns the arm with the highest utility with probability at least $1-\delta$ and has a round complexity of order

$$\mathcal{O}\left(\frac{1}{\min_{j\neq i^*}\Delta_{i^*,j}^2}\log\big(\frac{K}{\delta\,\min_{j\neq i^*}\Delta_{i^*,j}}\big)\right)$$

and a (pairwise) sample complexity of order $\mathcal{O}\left(\sum_{j\neq i^*}\frac{1}{\Delta_{i^*,j}^2}\log\big(\frac{K}{\delta\,\Delta_{i^*,j}}\big)\right)$. Further, it is shown that the algorithm is via a slight modification of its stopping criterion also an (ϵ,δ)-PAC learner with round complexity $\mathcal{O}\big(\frac{1}{\epsilon^2}\log\big(\frac{K}{\delta\,\epsilon}\big)\big)$ and sample complexity

$$\mathcal{O}\left(\sum_{j\notin \mathrm{CW}_\epsilon}\frac{\log\big(\frac{K}{\delta\,\Delta_{i^*,j}}\big)}{\Delta_{i^*,j}^2}+\frac{|\mathrm{CW}_\epsilon|}{\epsilon^2}\log\frac{K}{\epsilon\delta}\right), \tag{7}$$

where $\mathrm{CW}_\epsilon=\{i\in[K]\,|\,u_i\geq u_{i^*}-\epsilon\}$ is the set of ϵ-best arms, which is slightly different from the definition in Section 2.2.5.

3.2.7 Multisort

Maystre and Grossglauser (2017) address the ranking problem when comparison outcomes are generated from the Bradley-Terry (BT) (Bradley and Terry, 1952) probabilistic model with parameters $u=(u_1,\ldots,u_K)^\top\in\mathbb{R}_+^K$, which represent the utilities of the arms. Using the BT model, the probability that an arm a_i is preferred to a_j is given by

$$\mathbf{P}(a_i\succ a_j)=\frac{1}{1+\exp(-(u_i-u_j))}\ . \tag{8}$$

Thus, they end up with a utility-based dueling bandits problem with the logistic link function. The authors propose the Multisort algorithm for this setting, which essentially calls the QuickSort algorithm (Hoare, 1962) multiple times, stores the obtained (noisy) rankings and aggregates them to a final ranking by using Copeland's method (Copeland, 1951), in which the arms are increasingly sorted by their Copeland scores given by the total number of pairwise wins (cf. Section 4). In order to analyze the theoretical properties of Multisort, the authors assume that the utilities are well-separated, so that in particular a total order over the arms induced by the utilities exists. More specifically, it is assumed that the utility parameter u is generated by a Poisson point process. Then, it is shown that one call of QuickSort results in a ranking that is appropriately close to the underlying ranking of the arms with respect to Spearman's footrule distance given by $F(\sigma,\tau) = \sum_{i=1}^{n} |\sigma(i) - \tau(i)|$, where $\sigma(i)$ and $\tau(i)$ are the ranks of i according to the rankings σ and τ, respectively. With this, they can infer that using $\mathcal{O}(\log^5 K)$ calls of QuickSort within the Multisort algorithm results in a ranking that is close to the total order over the arms with respect to Spearman's footrule distance with high probability. As each call of QuickSort involves $\mathcal{O}(K \log K)$ many pairwise comparisons, the total number of comparisons made by Multisort is of the order $\mathcal{O}\left(K \log^6 K\right)$, while the distance of the final ranking to the underlying ranking of the arms is $o(K)$ in terms of F.

For practical purposes, the authors suggest to run QuickSort repeatedly for a budget of c pairwise comparisons, until the budget is exhausted, resulting in a set of c comparison pairs and their outcomes, while ignoring the produced rankings themselves. Eventually, the final ranking is built by sorting the ML estimates of the BT model parameters based on the set of all c pairwise comparison outcomes.

3.3 Regularity Through Statistical Models

Since one of the most general tasks in the realm of preference-based bandits is to elicit a ranking of the complete set of arms based on noisy (probabilistic) feedback, it is quite natural to establish a connection to statistical models of rank data (Marden, 1995).

The idea of relating preference-based bandits to rank data models has been put forward by Busa-Fekete et al. (2014a), who assume the underlying data-generating process to be given in the form of a probability distribution $\mathbf{P} : \mathbb{S}_K \to [0,1]$. Here, $\mathbb{S}_K$ is the set of all permutations of $[K]$ (the symmetric group of order K) or, via a natural bijection, the set of all rankings (total orders) of the K arms.

The probabilities for pairwise comparisons are then obtained as marginals of $\mathbf{P}$. More specifically, with $\mathbf{P}(\pi)$ the probability of observing the ranking π, the probability $q_{i,j}$ that a_i is preferred to a_j is obtained by summing over all rankings π in which a_i precedes a_j:

$$q_{i,j} := \mathbf{P}(a_i \succ a_j) = \sum_{\pi \in \mathcal{L}(r_j > r_i)} \mathbf{P}(\pi) \ , \tag{9}$$

where $\mathcal{L}(r_j > r_i) := \{\pi \in \mathbb{S}_K \,|\, \pi(j) > \pi(i)\}$ denotes the subset of permutations for which the rank of a_j is higher than the rank of a_i (smaller ranks indicate higher preference). In this setting, the learning problem essentially comes down to making inference about $\mathbf{P}$ based on samples in the form of pairwise comparisons.

So far, only two ranking models have been considered in the realm of PB-MAB problems, namely the Mallows model (Mallows, 1957) and the Plackett-Luce model (Plackett, 1975;

Luce, 1959). PB-MAB methods based on both models will be reviewed in the following subsections. Note that both models impose assumptions on the preference relation $\mathbf{Q}$ that are stronger than those for the utility-based dueling bandits (and thus also stronger than those in Section 3.1), as the pairwise probabilities in (9) of both models can be expressed in the form as stipulated by (5) for a suitable link function σ and utility function $u : \mathcal{A} \rightarrow \mathbb{R}$.

Under the assumption of a statistical model, the two targets of finding an optimal arm resp. ranking (cf. Section 2.2) can be formulated as follows.

– *Optimal arm*: The optimal arm i^* is the one having highest probability of being top-ranked:

$$i^* := \operatorname*{argmax}_{1 \le i \le K} \mathbf{E}_{\pi \sim \mathbf{P}} \, \mathbb{I}\{r_i = 1\} = \operatorname*{argmax}_{1 \le i \le K} \sum_{\pi \in \mathcal{L}(r_i = 1)} \mathbf{P}(\pi),$$

where $\mathcal{L}(r_i = 1) := \{\pi \in \mathbb{S}_K \,|\, \pi(i) = 1\}$ denotes the subset of permutations for which the rank of a_i is 1.

– *Ranking*: The target ranking π^* is the mode of the distribution:

$$\pi^* := \operatorname*{argmax}_{\pi \in \mathbb{S}_K} \mathbf{P}(\pi).$$

3.3.1 Mallows

Busa-Fekete et al. (2014a) assume the underlying probability distribution $\mathbf{P}$ to be a Mallows model (Mallows, 1957), one of the most well-known and widely used statistical models of rank data (Marden, 1995). The Mallows model or, more specifically, Mallows ϕ-distribution is a parameterized, distance-based probability distribution that belongs to the family of exponential distributions:

$$\mathbf{P}(\pi \,|\, \phi, \pi^*) = \frac{1}{Z(\phi)} \, \phi^{d(\pi, \pi^*)} \quad , \tag{10}$$

where ϕ and π^* are the parameters of the model: $\pi^* = (\pi_1^*, \ldots, \pi_K^*) \in \mathbb{S}_K$ is the location parameter (center ranking) specifying the position $\pi_i = \pi^*(i)$ of each arm a_i in the ranking, and $\phi \in (0, 1]$ is the spread parameter. Moreover, d is the Kendall distance on rankings, that is, the number of discordant pairs:

$$d(\pi, \pi^*) := \sum_{1 \le i < j \le K} \mathbb{I}\{\, (\pi_i - \pi_j)(\pi_i^* - \pi_j^*) < 0 \,\} \ ,$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function. The normalization factor in (10) can be written as

$$Z(\phi) = \sum_{\pi \in \mathbb{S}_K} \mathbf{P}(\pi \,|\, \phi, \pi^*) = \prod_{i=1}^{K-1} \sum_{j=0}^{i} \phi^j$$

and thus only depends on the spread (Fligner and Verducci, 1986). Note that, since $d(\pi, \pi^*) = 0$ is equivalent to $\pi = \pi^*$, the center ranking π^* is the mode of $\mathbf{P}(\cdot \,|\, \phi, \pi^*)$, that is, the most probable ranking according to the Mallows model.

In the case of Mallows, it is easy to see that $\pi_i < \pi_j$ implies $q_{i,j} > 1/2$ for any pair of items a_i and a_j. That is, the center ranking defines a total order on the set of arms. Moreover, as shown by Mallows (1957), the pairwise probabilities can be calculated analytically as functions of the model parameters ϕ and π^* as follows: Assume the Mallows model with parameters ϕ and π^*. Then, for any pair of items i and j such that $\pi_i < \pi_j$, the pairwise probability is given by $q_{i,j} = g(\pi_i, \pi_j, \phi)$, where

$$g(i, j, \phi) = h(j - i + 1, \phi) - h(j - i, \phi)$$

with $h(k, \phi) = k/(1 - \phi^k)$. As shown by this result, the Mallows model even induces a structural property over the arms like for the utility-based dueling bandits, so that the Mallows model is an even stricter assumption than the total order assumption[11]. Further, one can show that the "margin" around 1/2, i.e., $\min_{i \neq j} |1/2 - q_{i,j}|$, is relatively wide; more specifically, there is no pair (i, j) such that $q_{i,j} \in (\frac{\phi}{1+\phi}, \frac{1}{1+\phi})$. Moreover, the result also implies that $q_{i,j} - q_{i,k} = \mathcal{O}(\ell \phi^\ell)$ for arms a_i, a_j, a_k satisfying $\pi_i = \pi_j - \ell = \pi_k - \ell - 1$ with $1 < \ell$, and $q_{i,k} - q_{i,j} = \mathcal{O}(\ell \phi^\ell)$ for arms a_i, a_j, a_k satisfying $\pi_i = \pi_j + \ell = \pi_k + \ell + 1$ with $1 < \ell$. Therefore, deciding whether an arm a_j has higher or lower rank than a_i (with respect to π^*) is easier than selecting the preferred arm from two candidates a_j and a_k for which $j, k \neq i$.

For the problem of finding the optimal arm with high probability and a minimal sample complexity, Busa-Fekete et al. (2014a) propose the MallowsMPI algorithm, which is inspired by the algorithm for finding the largest element in an array. Any two arms a_i and a_j are compared until

$$1/2 \notin \left[\, \widehat{q}_{i,j} - c_{i,j}, \widehat{q}_{i,j} + c_{i,j} \,\right] \tag{11}$$

holds. This simple strategy finds the most preferred arm with probability at least $1 - \delta$ for a sample complexity that is of the form $\mathcal{O}\left(\frac{K}{\rho^2} \log \frac{K}{\delta\rho}\right)$, where $\rho = \frac{1-\phi}{1+\phi}$.

For the problem of finding the most probable ranking with high probability and a minimal sample complexity, a sampling strategy called MallowsMerge is proposed, which is based on the merge sort algorithm for selecting the arms to be compared. However, as in the case of finding the optimal arm, two arms a_i and a_j are not only compared once but until condition (11) holds. The MallowsMerge algorithm finds the most probable ranking, which coincides with the center ranking of the Mallows model, with a sample complexity of

$$\mathcal{O}\left(\frac{K \log_2 K}{\rho^2} \log \frac{K \log_2 K}{\delta\rho}\right) \quad ,$$

where $\rho = \frac{1-\phi}{1+\phi}$. The leading factor of the sample complexity of MallowsMerge differs from the one of MallowsMPI by a logarithmic factor, which is in line with the results in Section 3.1 for the exact sample complexity of best arm and top-K ranking.

Finally, the authors consider the KLD problem, which calls for producing a good estimate $\widehat{\mathbf{P}}$ of the distribution $\mathbf{P}$, that is, an estimate with small KL divergence:

$$\mathrm{KL}\left(\mathbf{P}, \widehat{\mathbf{P}}\right) = \sum_{\pi \in \mathbb{S}_K} \mathbf{P}(\pi) \log \frac{\mathbf{P}(\pi)}{\widehat{\mathbf{P}}(\pi)} < \epsilon.$$

11. Interestingly, the stochastic triangle inequality is not satisfied for the Mallows ϕ-model (Mallows, 1957).

The KLD problem turns out to be very hard for the case of Mallows, and even for small K, the sample complexity required for a good approximation of the underlying Mallows model is extremely high with respect to ϵ. In (Busa-Fekete et al., 2014a), the existence of a polynomial algorithm for this problem (under the assumption of the Mallows model) was left as an open question.

3.3.2 PLACKETT-LUCE

Szörényi et al. (2015a) assume that the underlying probability distribution is a Plackett-Luce (PL) model (Plackett, 1975; Luce, 1959). The PL model is parametrized by a vector $\theta = (\theta_1, \theta_2, \ldots, \theta_K) \in \mathbb{R}_+^K$, where each θ_i can be interpreted as the weight or "strength" of the arm a_i. The probability assigned by the PL model to a ranking represented by a permutation $\pi \in \mathbb{S}_K$ is given by

$$\mathbb{P}_\theta(\pi) = \prod_{i=1}^{K} \frac{\theta_{\pi^{-1}(i)}}{\theta_{\pi^{-1}(i)} + \theta_{\pi^{-1}(i+1)} + \ldots + \theta_{\pi^{-1}(K)}} \ , \tag{12}$$

where $\pi^{-1}(i)$ is the index of the item on position i. The product on the right-hand side of (12) is the probability of producing the ranking π in a *stagewise* process: First, the item on the first position is selected, then the item on the second position, and so forth. In each step, the probability of an item to be chosen next is proportional to its weight. Consequently, items with a higher weight tend to occupy higher positions. In particular, the most probable ranking (i.e., the mode of the PL distribution) is simply obtained by sorting the items in decreasing order of their weight:

$$\tau = \operatorname*{argmax}_{\pi \in \mathbb{S}_K} \mathbb{P}_\theta(\pi) = \operatorname*{argsort}_{k \in [K]} \{\theta_1, \ldots, \theta_K\} \ .$$

It is worth noting that the pairwise probabilities (9) for the PL model coincide with (8) by setting the parameters u_i of the Bradley-Terry model to $\log(\theta_i)$ for all $i \in \{1, \ldots, K\}$.

The authors consider two different targets of the learner, which are both meant to be achieved within an (ϵ, δ)-PAC learning setting. In the first problem, the goal is to find an ϵ-optimal arm, for which they devise the PLPAC algorithm with a sample complexity of $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{K}{\delta}\right)$. The second goal is to find an ϵ-optimal ranking, for which they propose the PLPAC-AMPR algorithm, whose sample complexity is of order $\mathcal{O}\left(\frac{K}{\epsilon^2} \log^2 \frac{K}{\delta}\right)$.

Both algorithms are based on a budgeted version of the QUICKSORT algorithm (Hoare, 1962), which reduces its quadratic worst-case complexity to the order $\mathcal{O}(K \log K)$, and in which the pairwise stability property is provably preserved (the pairwise marginals obtained from the distribution defined by the QUICKSORT algorithm coincide with the marginals of the PL distribution).

4. Learning from Non-coherent Pairwise Comparisons

The methods presented in the previous section essentially proceed from a given target, for example a ranking $\succ$ of all arms, which is considered as a "ground truth". The preference feedback in the form of (stochastic) pairwise comparisons provide information about this target and, consequently, should obey certain consistency or regularity properties. This is perhaps most explicitly expressed in Section 3.3, in which the $q_{i,j}$ are derived as marginals of a probability distribution on the set of all rankings, which can be seen as modeling a noisy observation of the ground truth given in the form of the center ranking.

Another way to look at the problem is to start from the pairwise preferences $\mathbf{Q}$ themselves, that is to say, to consider the pairwise probabilities $q_{i,j}$ as the ground truth. In tournaments in sports, for example, the $q_{i,j}$ may express the probabilities of one team a_i beating another one a_j. In this case, there is no underlying ground truth ranking from which these probabilities are derived. Instead, it is just the other way around: A ranking is derived from the pairwise comparisons. Moreover, there is no reason for why the $q_{i,j}$ should be coherent in the sense that a natural total ordering of the arms or a Condorcet winner exists. In particular, preferential cycles and violations of transitivity are commonly observed in practice, so that alternative concepts for the definition of a target, such as an optimal arm or a ranking, are needed. In the following sections, we provide an overview of approaches that adopt this point of view, thereby providing the missing supplement to the taxonomy of PB-MAB algorithms (cf. Figure 1)

4.1 Alternative Target Concepts

The challenge caused by non-coherence such as cycles or violations of transitivity is faced by *ranking procedures*, which have been studied quite intensely in operations research and decision theory (Moulin, 1988; Chevaleyre et al., 2007). A ranking procedure $\mathcal{R}$ turns $\mathbf{Q}$ into a complete preorder relation $\succ^{\mathcal{R}}$ of the arms under consideration. Thus, another way to pose the preference-based MAB problem is to instantiate $\succ$ with $\succ^{\mathcal{R}}$ as the target for prediction—the connection between $\mathbf{Q}$ and $\succ$ is then established by the ranking procedure $\mathcal{R}$, which of course needs to be given as part of the problem specification.

Formally, a ranking procedure $\mathcal{R}$ is a map $[0,1]^{K\times K} \to \mathcal{C}_K$, where $\mathcal{C}_K$ denotes the set of complete preorders on the set of arms. We denote the complete preorder produced by the ranking procedure $\mathcal{R}$ on the basis of $\mathbf{Q}$ by $\succ^{\mathcal{R}}_{\mathbf{Q}}$, or simply by $\succ^{\mathcal{R}}$ if $\mathbf{Q}$ is clear from the context. Below we present some of the most common instantiations of the ranking procedure $\mathcal{R}$.

- Copeland's ranking (CO) is defined as follows (Moulin, 1988): $a_i \succ^{\mathrm{CO}} a_j$ if and only if $d_i > d_j$, where $d_i := \#\{k \in [K] \,|\, 1/2 < q_{i,k}\}$ is the Copeland score of a_i. The interpretation of this relation is very simple: An arm a_i is preferred to a_j whenever a_i "beats" more arms than a_j does.
- The sum of expectations (SE) (or Borda) ranking is a "soft" version of CO: $a_i \succ^{\mathrm{SE}} a_j$ if and only if
$$q_i = \frac{1}{K-1}\sum_{k\neq i} q_{i,k} > \frac{1}{K-1}\sum_{k\neq j} q_{j,k} = q_j \quad . \tag{13}$$
- The idea of the random walk (RW) ranking is to handle the matrix $\mathbf{Q}$ as a transition matrix of a Markov chain and order the arms based on its stationary distribution. More precisely, RW first transforms $\mathbf{Q}$ into the stochastic matrix $\mathbf{S} = [s_{i,j}]_{K\times K}$, where $s_{i,j} = q_{i,j}/\sum_{\ell=1}^{K} q_{\ell,i}$. Then, it determines the stationary distribution $(v_1,\ldots,v_K)$ for this matrix (i.e., the normalized eigenvector corresponding to the largest eigenvalue

1). Finally, the arms are sorted according to these probabilities: $a_i \succ^{\text{RW}} a_j$ iff $v_i > v_j$. The RW ranking is directly motivated by the PageRank algorithm (Brin and Page, 1998), which has been well studied in social choice theory (Cohen et al., 1999; Brandt and Fischer, 2007) and rank aggregation (Negahban et al., 2012), and which is widely used in many application fields (Brin and Page, 1998; Kocsor et al., 2008).

Unlike the total order over arms, which may not exist for a specific preference relation $\mathbf{Q}$, all the just mentioned notions of a target ranking are defined for any preference relation $\mathbf{Q}$. However, in case the total ordering over the arms exists, neither the random walk ranking nor the Borda ranking will necessarily coincide with the former, whereas the Copeland ranking provably does. Indeed, consider the preference relation

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0.55 & 0.55 \\ 0.45 & 0.5 & 1 \\ 0.45 & 0 & 0.5 \end{pmatrix}. \tag{14}$$

Then, there is a total ordering given by $a_1 \succ a_2 \succ a_3$, but the Borda ranking as well as the random walk ranking are $a_2 \succ^{\text{SE}} a_1 \succ^{\text{SE}} a_3$, as $q_1 = 0.55$, $q_2 = 0.725$, and $q_3 = 0.275$. Furthermore, the random walk ranking coincides with the Borda ranking, that is $a_2 \succ^{\text{RW}} a_1 \succ^{\text{RW}} a_3$, since the stationary distribution of the induced stochastic matrix of $\mathbf{Q}$ is $(v_1, v_2, v_3) \approx (0.4107, 0.4147, 0.1746)$.

Groves and Branke (2019) provide a sufficient condition for the coherence of the Borda ranking and the total order over arms in the case $\mathbf{Q}$ fulfills strong stochastic transitivity. In particular, if for any two distinct arms $a_i, a_j \in \mathcal{A}$, there exists some arm a_k such that $\Delta_{i,k} \neq \Delta_{j,k}$, then the Borda ranking is the same as the total order. Moreover, if the low noise assumption holds, then the Borda ranking as well as the random walk ranking coincide with the total ordering.

The alternative notions for a target ranking also allow one to define alternatives for the Condorcet winner, which, just like the total order of the arms, may not exist for some preference relations $\mathbf{Q}$. In particular, the following definitions of optimal arms are considered in the literature.

- *Copeland winner*: The Copeland set $\text{CP}(\mathbf{Q})$ is defined as the set of arms in $[K]$ with highest Copeland score, i.e., $\text{CP}(\mathbf{Q}) = \{i \in [K] \,|\, i \in \text{argmax}_{j \in [K]} d_j\}$. In other words, the Copeland set consists of arms having the top position in the preorder defined by $\succ^{\text{CO}}$. Each arm in the Copeland set has the property that it beats the maximal number of other arms, and hence is called a Copeland winner. In contrast to the Condorcet winner, a Copeland winner can be beaten by another arm. The cardinality of such arms, i.e., the number of arms by which a Copeland winner is beaten, will be denoted by L_C.

- *Borda winner/random walk winner*: Similarly as for the Copeland winner, it is possible to define the Borda winner resp. the random walk winner as the arms having the top position according to the preorder on the alternatives induced by $\succ^{\text{SE}}$ resp. $\succ^{\text{RW}}$. Just like in the case of the Copeland winner, these alternative notions of optimality of an arm do not exclude the possibility of being beaten by another arm.

- *von Neumann winner*: Inspired by the game-theoretical point of view on the dueling bandits problem (cf. Section 2.4.1), one can define a zero-sum game matrix $P = 2\mathbf{Q}-1$. By setting the outcome of the game to $+1$ if a_i wins and -1 if a_j wins, so that the entry of P at position (i,j) is the expected outcome of a duel between the arms a_i and a_j. This matrix P is skew-symmetric and specifies a zero-sum game, so that von Neumann's minmax theorem (von Neumann, 1928; Owen, 1982) implies the existence of a probability vector $\mathcal{P} \in \Delta^{(K-1)}$ which is a maxmin strategy for the game described by P[12]. This probability vector $\mathcal{P}$ is called the von Neumann winner. In contrast to the alternative notions of optimality, it thus represents a random strategy instead of a pure strategy (a single arm or a set of arms) as an optimality concept. The von Neumann winner specifically satisfies $\sum_{i=1}^K \mathcal{P}(i)\, q_{i,j} \geq \sum_{i=1}^K \mathcal{P}(i)\, q_{j,i}$ for any $j \in [K]$, so that it guarantees a higher probability of winning by choosing an arm according to $\mathcal{P}$ than loosing against any arm a_j.

The relationship between the alternative optimality concepts for arms and the Condorcet winner resembles their ranking counterparts. If the Condorcet winner exists, the Copeland set reduces to a singleton set with the Condorcet winner as its only element. The Borda resp. random walk winner, on the other hand, may differ from the Condorcet winner (consider the preference relation in (14)). However, if the low noise model assumption is satisfied, the Borda and random walk winner are both unique and equal the Condorcet winner. Regarding the von Neumann winner $\mathcal{P}$, it is well known that it coincides with the Condorcet winner a^* if the latter exists, in the sense that $\mathcal{P}$ is the Dirac measure on the singleton set $\{a^*\}$. In cases where a Condorcet winner does not exist, the von Neumann winner will generally differ from the Copeland, Borda, and random walk winner (cf. Appendix B in Dudík et al. (2015)).

4.2 Algorithms for Non-coherent Preference Relations

In this section, we survey existing methods for PB-MAB problems that do not proceed from specific structural assumptions on $\mathbf{Q}$. Instead, the targets of the learner could be derived from a possibly non-coherent preference relation. In Tables 5—7, we summarize the approaches according to their algorithm class, the targets and goals considered, as well as their theoretical guarantees.

4.2.1 Copeland Confidence Bound

Zoghi et al. (2015a) introduce what is currently known in the literature as the Copeland dueling bandit problem, namely the task of regret minimization with the cumulative regret as in (2) and the instantaneous regret suffered by the learner when comparing $a_{i(t)}$ and $a_{j(t)}$ at time t set to $2\mathrm{cp}(a_C) - \mathrm{cp}(a_{i(t)}) - \mathrm{cp}(a_{j(t)})$, where $a_C \in \mathrm{CP}(\mathbf{Q})$ is a Copeland winner of the underlying preference relation $\mathbf{Q}$ and $\mathrm{cp}(a_i) = \frac{d_i}{K-1}$ is the normalized Copeland score of an arm $a_i \in \mathcal{A}$. It is worth noting that in case $\mathbf{Q}$ has a Condorcet winner, all theoretical guarantees in the form of regret bounds derived for algorithms on a Copeland dueling bandit problem can be transferred to regret bounds on the cumulative regret in (2) by a multiplicative constant. In particular, algorithms for Copeland dueling bandits can be

12. $\Delta^{(K-1)}$ is the $K-1$ probability simplex.

Algorithm	Algorithm class	Target(s) and goal(s) of learner	Theoretical guarantee(s)
1. Copeland confidence bound 2. Scalable Copeland bandits (Section 4.2.1)	1. Generalization-based (UCB), tournament 2. Generalization-based (Explore-then-exploit), Reduction-based	High probability and expected regret minimization for finding the Copeland winner	1. $\mathcal{O}\left(\frac{K^2+K(\lvert \mathrm{CP}(\mathbf{Q})\rvert+L_C)\log T}{\min_{i\neq j}\Delta_{i,j}^2}\right)$ 2. $\mathcal{O}\left(\frac{K(\log(K)+L_C)\log(T)}{\min_{i\neq j}\Delta_{i,j}^2}\right)$
Copeland winners relative minimum empirical divergence (Section 4.2.2)	Generalization-based (Explore-then-exploit, DMED)	Expected regret minimization for finding the Copeland winner	$\mathcal{O}\left(\min_{k\in \mathrm{CP}(\mathbf{Q})} C_k^*(\mathbf{Q})\log(T)\right)$ $+\, o(\log(T))$ resp. $\mathcal{O}\left(\frac{K\, L_C\, \log T}{\min_{i\neq j}\Delta_{i,j}^2}\right)$
Double Thompson sampling (Section 4.2.3)	Generalization (Thompson sampling and upper/lower confidence bounds)	Expected regret minimization for finding the Copeland winner	$\mathcal{O}\left(\frac{K^2\log\log T+K(\lvert \mathrm{CP}(\mathbf{Q})\rvert+L_C)\log T}{\min_{i\neq j}\Delta_{i,j}^2}\right)$ $+\,\mathcal{O}(K^3)$
Sparse sparring (Section 4.2.4)	Reduction-based, generalization (successive elimination)	High probability regret minimization for the von Neumann winner	See (15)
General tournament solutions (Section 4.2.5)	Generalization (UCB)	High probability regret minimization for finding the Copeland set, the top cycle, uncovered set, and Banks set	$\mathcal{O}\left(K^2+\frac{\log T}{\Delta^{(TS)}}\right)$, where $\Delta^{(TS)}$ is a target specific complexity term

Table 5: Approaches for regret minimization tasks in the dueling bandits problem with non-coherent pairwise comparisons.

Algorithm	Algorithm class	Target(s) and goal(s) of learner	Theoretical guarantee(s)
PAC rank elicitation (Section 4.2.6)	Generalization-based (racing/successive elimination)	(ϵ,δ)-PAC learning of Copeland and Borda rankings with the NDP and MRD measures for near optimality	$\mathcal{O}\left(R(\epsilon,\mathbf{Q})\log\frac{R(\epsilon,\mathbf{Q})}{\delta}\right)$ with a function R depending on ϵ and $\mathbf{Q}$.
Borda-ranking (Section 4.2.7)	Generalization (explore-then-exploit), reduction-based	(ϵ,δ)-PAC learning for finding 1. Borda winner 2. Borda ranking	1. $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{1}{\delta}\right)$ 2. $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{K}{\delta}\right)$
Round-Efficient Dueling Bandits (Section 4.2.8)	Tournament	(ϵ,δ)-PAC sample and round complexity minimization for Borda winner	Round complexity: $\mathcal{O}\left(\frac{1}{\epsilon^2}\log\frac{K}{\epsilon\delta}\right)$ Sample complexity: $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{K}{\epsilon\delta}\right)$
Hamming-LUCB (Section 4.2.9)	Generalization (lower-upper confidence bound)	(ϵ,δ)-PAC learning of the top-k Borda arms (and their complementary set) w.r.t. the Hamming distance	See (16)

Table 6: Approaches for (ϵ,δ)-PAC settings of the dueling bandits problem with non-coherent pairwise comparisons.

Algorithm	**Algorithm class**	**Target(s) and goal(s) of learner**	**Theoretical guarantee(s)**
Preference-based racing (Section 4.2.10)	Generalization-based (racing/successive elimination)	Exact sample complexity minimization for top-k ranking w.r.t. Copeland, Borda, and Random walk ranking	$\mathcal{O}\left(\sum_{1\le i<j\le K}\frac{\log\frac{K}{\delta}}{(\Delta_{i,j})^2}\right)$
Voting bandits (Section 4.2.11)	Generic zooming algorithm	Exact sample complexity minimization for finding Copeland and Borda winners	$\mathcal{O}\left(\sum_{1\le i<j\le K}\frac{\log\frac{K}{\delta\Delta_{i,j}}}{\Delta_{i,j}^2}\right)$
Successive elimination (Section 4.2.12)	Reduction-based	Exact sample complexity for Borda winner	See (17)
Pairwise Optimal Computing Budget Allocation & Pairwise Knowledge Gradient (Section 4.2.13)	Generalization (Bayesian Expected Improvement Sampling)	Asymptotic optimality for top-k arms with respect to the Borda ranking	Consistency of the algorithm, but no bound on the sample complexity
Active ranking (Section 4.2.14)	Generalization (racing/successive elimination)	Exact sample complexity minimization for Borda coarse ranking	See (18)

Table 7: Approaches for exact sample complexity tasks of the dueling bandits problem with non-coherent pairwise comparisons.

used for the regret minimization task considered in Section 3.1 as well, since the former is a more general problem scenario than the latter.

For the Copeland dueling bandit problem, Zoghi et al. (2015a) suggest two algorithms: First, the Copeland confidence bound (CCB) algorithm, which is suited for learning problems with a small number of arms and borrows some ideas underlying the RUCB algorithm (cf. Section 3.1.3). Second, the scalable Copeland bandits (SCB) algorithm, which can deal more efficiently with learning problems having a large number of available arms and makes use of the idea underlying the KL-UCB algorithm for value-based MAB problems (Garivier and Cappé, 2011; Cappé et al., 2013) in order to find a Copeland winner arm.

CCB is making use of the principle underlying UCB, i.e., the optimism in the face of uncertainty principle, and additionally its pessimistic counterpart by considering upper confidence and lower confidence estimates of the pairwise preference probabilities simultaneously. More specifically, in each time step the optimistic/pessimistic Copeland scores are computed based on the upper/lower confidence estimates. These scores are then used to (i) determine a set of optimistic Copeland winners from which, except for a probabilistic exploration, the first arm of the pair is sampled (almost) uniformly at random[13], (ii) eliminate all likely non-Copeland winner arms, and (iii) create a set consisting of tough competitors for each arm. The key idea underlying the choice of the second arm is then to use an arm which has the highest potential to refute that the first chosen arm is a Copeland winner. More specifically, among all arms having a lower confidence estimate for the pairwise preference probability against the first arm at most 1/2, the one with the lower confidence estimate closest to 1/2 is chosen (possibly breaking ties). Thus, if the lower confidence estimate for

13. The "almost" is due to a slight bias given to arms in the optimistic Copeland winners set, which have shown to be good candidates over the time.

the pairwise preference probability against the first arm exceeds 1/2, then the upper confidence estimate for the pairwise preference probability of the first arm being preferred over the second arm falls below 1/2, which in turn lowers the first arm's optimistic Copeland score. In order to introduce additional "exploration" in this regard, the second arm with the above property is picked with probability 1/2, respectively, from the set of all tough competitors of the first arm and among all arms. Assuming that there are no ties, i.e., $q_{i,j} \neq 1/2$ for all pairs (a_i, a_j), $i \neq j$, the authors provide for CCB both high probability and expected (Copeland) regret bounds of the form $\mathcal{O}\left(\frac{K^2+K(|\mathrm{CP}(\mathbf{Q})|+L_C)\log T}{\min_{i\neq j}\Delta_{i,j}^2}\right)$. In fact, the shown regret bound is stricter, because the gap term $\min_{i\neq j}\Delta_{i,j}^2$ can be replaced by a slightly larger gap term.

SCB is an explore-then-exploit algorithm, which is made horizonless by using the doubling or squaring trick (Cesa-Bianchi and Lugosi, 2006). During its exploration phase within one of the doubling periods, the algorithm uses a subroutine, which is a variation of the KL-UCB algorithm for a best arm identification problem of a value-based MAB setting with zero–one rewards. In order to specify a reasonable reward mechanism for each arm, the algorithm simulates, each time a value-based reward of the pulled arm a_i is expected, the following process: First, another arm $a_j \in \mathcal{A}\backslash\{a_i\}$ is chosen uniformly at random, then a_i and a_j are dueled with each other a specific period-dependent number of times, resulting in a reward of 1 in case a_i has won the majority of the duels and a reward of 0 otherwise. It can be shown that the best arm(s) based on these induced reward distributions correspond(s) to the Copeland winner(s). After the KL-UCB variant has identified the best arm using this reward simulation process, SCB starts the exploitation phase of the doubling period by dueling the prior arm against itself (cf. "full commitment" in Section 2.3.1). If the available time budget of the corresponding doubling period is not sufficient to let the exploration phase come to an end, no exploitation phase is conducted in this period and instead the exploration phase of the next period is started. Once again assuming that the preference relation has no ties, SCB achieves an expected (Copeland) regret bound of $\mathcal{O}\left(\frac{K(\log(K)+L_C)\log(T)}{\min_{i\neq j}\Delta_{i,j}^2}\right)$, which has a more favorable dependency on the number of arms K than the regret bound of CCB and does not depend on the number of Copeland winners $\mathrm{CP}(\mathbf{Q})$.

4.2.2 Copeland Winners Relative Minimum Empirical Divergence

Komiyama et al. (2016) consider the Copeland dueling bandit problem as in Zoghi et al. (2015a), but modify the instantaneous regret by a multiplicative constant factor. For this problem scenario, they propose the Copeland Winners Relative Minimum Empirical Divergence (CW-RMED) algorithm, which, just like the relative minimum empirical divergence algorithm (cf. Section 3.1.7), is inspired by the DMED algorithm (Honda and Takemura, 2010) for the value-based MAB problem. Thus, the design idea underlying the algorithm is guided by the lower bound for the respective bandit problem. To this end, the authors derive an asymptotic regret lower bound for any no-regret algorithm under the assumption that there are no ties in the preference relation $\mathbf{Q}$, which is of the form $\Omega\left(\min_{k\in\mathrm{CP}(\mathbf{Q})} C_k^*(\mathbf{Q})\log(T)\right)$, where $C_k^*(\mathbf{Q})$ is a complexity term coming from $\mathbf{Q}$ and depending on the Kullback-Leibler divergence of Bernoulli random variables with success probability equal to the pairwise probabilities and $1/2$ in the same spirit as in Section 3.1.7. This complexity term $C_k^*(\mathbf{Q})$ has a natural interpretation coming from the proof of the lower bound, namely $C_k^*(\mathbf{Q})\log(T)$ represents the smallest possible cumulative regret to ensure that $k \in \mathrm{CP}(\mathbf{Q})$, i.e., arm a_k is a Copeland winner.

As a consequence, CW-RMED revolves around the natural estimate of the complexity term by using the current estimates of the entries of the preference relation. Formally, $C_k^*(\hat{\mathbf{Q}}_t)$ with $\hat{\mathbf{Q}}_t = [\widehat{q}_{i,j}^{\,t}]_{1 \le i,j \le K}$ and $i_t^* \in \mathrm{argmin}_{i \in [K]} C_i^*(\hat{\mathbf{Q}}_t)$ are used, where the latter set specifies the candidates for the likeliest Copeland winners. After making sure that every pair of arms has been dueled a sufficient time-dependent number of times and all pairwise preference estimates are far enough away from $1/2$, the CW-RMED checks whether the current these estimates provide enough evidence in favor of some i_t^* being indeed a Copeland winner. If this is the case, the corresponding arm is dueled against itself (cf. "full commitment" in Section 2.3.1) in the upcoming time steps until the confidence of the belief is doubtful, in which case the pair of arms with the highest chance to regain confidence is dueled.

Because the computation necessary to determine the latter pair is in fact cumbersome, the authors propose the Efficient CW-RMED (ECW-RMED) algorithm, which gets rid of the latter computational problem by slightly modifying the target complexity term C_k^* to a reasonable surrogate $\tilde{C}_k^*$. As a consequence, the exploratory behavior of ECW-RMED is different from CW-RMED and leads to a (Copeland) regret bound of the former of $\mathcal{O}\left(\min_{k \in \mathrm{CP}(\mathbf{Q})} \tilde{C}_k^*(\mathbf{Q})\log(T)\right) + o(\log(T))$, while the latter has provably an expected regret bound of $\mathcal{O}\left(\min_{k \in \mathrm{CP}(\mathbf{Q})} C_k^*(\mathbf{Q})\log(T)\right) + o(\log(T))$. However, both complexity terms are the same if there is more than one Copeland winner, i.e., $|\mathrm{CP}(\mathbf{Q})| \ge 2$.

Finally, the authors relate the resulting regret bound of ECW-RMED to CCB, showing improvements regarding the multiplicative constants. More specifically, they show that, for the sake of comparison, the latter bounds can be further bounded by $\mathcal{O}\left(\frac{K\, L_C\, \log T}{\min_{i \neq j} \Delta_{i,j}^2}\right)$, revealing that the dependency on $\mathrm{CP}(\mathbf{Q})$ as a multiplicative factor in the (Copeland) regret can indeed be eliminated as it is the case for SCB.

4.2.3 Double Thompson Sampling

For the Copeland dueling bandit problem, Wu and Liu (2016) suggest the Double Thompson Sampling (D-TS) algorithm as well as an enhanced version of it (D-TS$^+$). Both algorithms rely on the well-known Thompson sampling (Thompson, 1933; Agrawal and Goyal, 2013; Kaufmann et al., 2012; Chapelle and Li, 2011) algorithm for value-based MAB problems in the sense that Beta posterior distributions over the entries of the preference relation are maintained as in the RCS algorithm (cf. Section 3.1.6) and are consequently used to suggest arms for the duel at one iteration step. Additionally, both algorithms borrow some ideas underlying the design of the CCB algorithm for choosing a maximum informative second arm to possibly refute that the first chosen arm is a Copeland winner. To be more precise, D-TS first samples a preference relation $\tilde{\mathbf{Q}} \in [0,1]^{K \times K}$ in each iteration[14] according to the current Beta posterior distributions and chooses for the first arm the one with the largest

14. Strictly speaking, the preference relation $\tilde{\mathbf{Q}}$ depends on the iteration step t, as the Beta posterior distributions do, but we suppress this dependency here in the notation for sake of convenience.

Copeland score on $\tilde{\mathbf{Q}}$ among all arms currently having the largest optimistic Copeland score (based on upper confidence estimates), while possible ties are broken randomly. Then, for choosing the second arm, D-TS uses once again the current Beta posterior distributions to sample for any other arm the pairwise preference probability for being preferred over the first arm, and chooses the arm with the largest sampled pairwise preference probability among all arms having a lower confidence estimate for the pairwise preference probability against the first arm of at most 1/2 (possibly breaking ties). Thus, the mechanism in choosing the second arm is similar to CCB, but differs due to the usage of the sampled pairwise preference probabilities instead of the upper confidence estimates.

The enhanced version D-TS$^+$ differs from D-TS in the possible tie breaking for choosing the first arm. More specifically, D-TS$^+$ additionally estimates the cumulative Copeland regret of an arm dueled against any other arm through the history of the sampled preference relations $\tilde{\mathbf{Q}}$ and, in the case of ties, chooses among all optimistic Copeland winners the one with the lowest estimated regret.

It is shown that D-TS achieves an expected cumulative (Copeland) regret bound of $\mathcal{O}\left(\frac{K^2 \log T}{\min_{i \neq j} \Delta_{i,j}^2} + K^2\right)$, provided there are no ties in the preference relation. Under an additional assumption on the pairwise probabilities of non-Copeland winner arms, the expected cumulative (Copeland) regret bound of D-TS is refined to

$$\mathcal{O}\left(\frac{K^2 \log\log T + K(|\mathrm{CP}(\mathbf{Q})| + L_C) \log T}{\min_{i \neq j} \Delta_{i,j}^2} + K^3\right).$$

The latter bounds are also shown to be valid for D-TS$^+$.

4.2.4 Sparse Sparring

Balsubramani et al. (2016) adopt the game-theoretic view of dueling bandits, as described in the definition of the von Neumann winner. The authors aim for algorithms the performance of which approach the performance of the von Neumann winner. To this end, they consider the cumulative regret up to time T as $\max_{k \in K} \sum_{t=1}^{T} (P_{k,i(t)} + P_{k,j(t)})/2$, where P is the zero-sum game matrix of $\mathbf{Q}$ (see the beginning of the section). This notion of regret was initially introduced by Dudík et al. (2015), which will be discussed in another context in Section 5.2.

The construction idea of the proposed Sparse Sparring (SPAR2) algorithm is guided by the observation that the von Neumann winner is usually sparse in the sense that only $s \ll K$ many of its entries are non-zero, which can be interpreted as having only a small set of good arms, while most of the arms are far from being optimal.

SPAR2 makes use of the sparring idea suggested by Ailon et al. (2014b) (cf. Section 5.1). In a nutshell, two algorithms for the value-based MAB problem are used to determine the pair of arms for the duel. One of the algorithms suggests the first arm, the other the second arm, and after having conducted the duel, the algorithm which suggested the winning arm obtains a reward of 1, the other a reward of -1. The key idea of SPAR2 is to use this sparring approach with two independent copies of the Exp3 algorithm (Auer et al., 2002b) and simultaneously maintain confidence regions around the empirical estimate of P in order to successively reduce the set of suggestible arms for the Exp3 algorithms until only the arms in the sparse support of the actual von Neumann winner arm remain. Roughly speaking, the latter reduction is realized by constantly checking the conformity of the supports of all von Neumann winners extractable from the confidence regions around the empirical estimate of P and eliminating an arm as soon as it is the first time not within the support of all these von Neumann winners. Because this approach is computationally inefficient, the authors propose a more efficient workaround, for which results on the stability of the von Neumann support are derived to introduce more conservative yet practicable conditions for monitoring the conformity of the supports.

Under the assumption that the von Neumann winner is unique and has s many non-zero probabilities, it is shown that SPAR2 enjoys a cumulative regret bound of

$$\min\Big\{\tilde{\mathcal{O}}(\sqrt{sT\log(s/\delta)} + C(P)\log(1/\delta)^2)\,,\, \mathcal{O}(\sqrt{KT\log(K/\delta)})\Big\} \tag{15}$$

with probability at least $1-\delta$, where $C(P)$ is some constant depending only on the zero-sum game matrix of $\mathbf{Q}$. This improves upon the regret bound of $\tilde{\mathcal{O}}(\sqrt{KT})$, which holds for the straightforward approach of simply using the two independent copies of the Exp3 algorithm.

4.2.5 General Tournament Solutions

Ramamohan et al. (2016) consider general tournament solutions from social choice theory (Brandt et al., 2016) as sets of winning arms. To this end, they relate a preference relation Q with a tournament graph by means of $\mathcal{T}_Q = ([K], E_Q)$ with $[K]$ as the set of vertices and $E_Q = \{(i,j) : q_{i,j} > 1/2\}$ as the set of edges. Moreover, a sub-tournament $\mathcal{T} = (V, E)$ of $\mathcal{T}_Q$ with $V \subseteq [K]$ and $E \subseteq E_Q$ is called *maximal acyclic*, if it is acyclic and no other sub-tournament of $\mathcal{T}$ is acyclic. With this, alternative general tournament solutions besides the Copeland set and the Condorcet winner can be defined as follows:

- the *top cycle* (or also known as the *Smith-set*) defined as the smallest set $W \subseteq [K]$ for which $q_{i,j} > 1/2$ for all $i \in W$ and $j \notin W$ holds,

- the *uncovered set* defined as the set of all arms that are not covered by any other arm, where an arm a_i is *covered* by an arm a_j if $q_{i,j} > 1/2$ and for any $k \in [K]\backslash\{i,j\}$ it holds that $q_{j,k} > 1/2$ implies $q_{i,k} > 1/2$,

- and the *Banks set* defined as the set consisting of the maximal elements of all maximal acyclic sub-tournaments of $\mathcal{T}_Q$.

It is worth noting that in case a Condorcet winner exists, all these alternative tournament solutions sets coincide and are singleton sets consisting of the Condorcet winner.

The authors measure the regret per time relative to the tournament solution of interest, i.e., a suitable notion of individual regret of an arm with respect to a tournament solution is defined and used to define the common pairwise regret. In order to minimize this notion of regret, a generic upper confidence bound (UCB) style dueling bandits algorithm is developed, UCB-TS, which can be instantiated for a specific tournament solution.

The generic algorithm maintains upper confidence bounds on the pairwise preference probabilities and updates these as well as all necessary statistics after observing the outcome of a duel. The composition of the duel is determined by a selection procedure especially designed for a specific tournament solution. For each tournament solution, one suitable instantiation of the selection procedure is proposed, respectively. All selection procedures are conceptionally similar to RUCB (cf. Section 3.1.3) or CCB (cf. Section 4.2.1): The first arm is chosen to be a candidate for the respective tournament solution based on the upper confidence bounds, while the second arm is chosen as the one having the highest potential to refute the validity that the first arm is indeed a (respective) tournament solution.

For each tournament solution, say TS, the specific instantiation of the generic algorithm is theoretically analyzed by showing high probability bounds on the regret of the form $\mathcal{O}\left(K^2 + \frac{\log T}{\Delta^{(\mathrm{TS})}}\right)$, where $\Delta^{(\mathrm{TS})}$ is a complexity term representing the difficulty of a specific tournament solution comparable to the complexity term $\sum_{j \neq i^*} \Delta_{i^*,j}$ occurring in the regret bound of RUCB (cf. Section 3.1.3), which assumes the existence of a Condorcet winner a_{i^*}. In the worst-case, these bounds are of the order $\mathcal{O}\left(K^2 \log T\right)$, and it remains an open question whether a lower bound of the same order can be shown.

4.2.6 PAC Rank Elicitation

Busa-Fekete et al. (2014b) were one of the first authors considering the goal of finding a ranking that is close to the ranking produced by some ranking procedure $\mathcal{R}$ in an (ϵ, δ)-PAC learning scenario. To formalize the notion of closeness of the (predicted) permutation τ with a (target) order $\succ$, two distance measures are considered. First, the *number of discordant pairs* (NDP), which is closely connected to Kendall's rank correlation (Kendall, 1955), and can be expressed as

$$d_{\mathcal{K}}(\tau, \succ) = \sum_{i=1}^{K} \sum_{j \neq i} \mathbb{I}\{\tau_j < \tau_i\} \mathbb{I}\{a_i \succ a_j\} \ ,$$

where τ_i denotes the rank of arm a_i in the permutation τ. The second is the *maximum rank difference* (MRD) defined as the maximum difference between the rank of an arm a_i according to τ and $\succ$, respectively. More specifically, since $\succ$ is a partial but not necessarily total order, τ is compared to the set $\mathcal{L}^{\succ}$ of its linear extensions[15]:

$$d_{\mathcal{M}}(\tau, \succ) = \min_{\tau' \in \mathcal{L}^{\succ}} \max_{1 \leq i \leq K} |\tau_i - \tau'_i| \ .$$

The authors point out the fact that, regarding the induced order relation $\succ^{\mathrm{CO}}$, ranking procedures such as Copeland might be strongly influenced by a minimal change of a value $q_{i,j} \approx \frac{1}{2}$. Consequently, the number of samples needed to assure (with high probability) a certain approximation quality ϵ may become arbitrarily large. A similar problem arises for $\succ^{\mathrm{SE}}$ as a target order if some of the individual Borda scores q_i are very close or equal to each other.

As a practical (yet meaningful) solution to this problem, the authors propose to impose stronger requirements on the order of relations such as $\succ^{\mathrm{CO}}$ and $\succ^{\mathrm{SE}}$ in order to make these more "partial". To this end, let $d_i^* := \#\left\{k \,|\, 1/2 + \gamma < q_{i,k}, i \neq k\right\}$ denote the number of arms that are preferred over a_i with a margin $\gamma > 0$, and let $s_i^* := \#\left\{k : |1/2 - q_{i,k}| \leq \gamma,\, i \neq k\right\}$. Then, the γ-insensitive Copeland relation is defined as follows: $a_i \succ^{\mathrm{CO}_\gamma} a_j$ if and only if

15. $\tau \in \mathcal{L}^{\succ}$ iff $\forall\, i, j \in [K] : (a_i \succ a_j) \Rightarrow (\tau_i < \tau_j)$

$d_i^* + s_i^* > d_j^*$. Likewise, in the case of the Borda ranking $\succ^{\mathrm{SE}}$, small differences of the q_i are neglected and the γ-insensitive sum of expectations relation is defined as follows: $a_i \succ^{\mathrm{SE}_\gamma} a_j$ if and only if $q_i + \gamma > q_j$.

These γ-insensitive extensions are interval (and hence partial) orders, that is, they are obtained by characterizing each arm a_i by the interval $[d_i^*, d_i^* + s_i^*]$ and sorting intervals according to $[a,b] \succ [a',b']$ iff $b > a'$. It is readily shown that $\succ^{\mathrm{CO}_\gamma} \subseteq \succ^{\mathrm{CO}_{\gamma'}} \subseteq \succ^{\mathrm{CO}}$ for $\gamma > \gamma'$, with equality $\succ^{\mathrm{CO}_0} \equiv \succ^{\mathrm{CO}}$ if $q_{i,j} \neq 1/2$ for all $i \neq j \in [K]$ (and similarly for the Borda ranking). The parameter γ controls the strictness of the order relations, and thereby the difficulty of the rank elicitation task.

The authors propose four different methods for the two γ-sensitive ranking procedures, along with the two distance measures described above. Each algorithm calculates a surrogate ranking based on the empirical estimate of the preference relation whose distance can be upper-bounded again based on some statistics of the empirical estimates of preference. The sampling is carried out in a greedy manner in every case, in the sense that those arms are compared which are supposed to result in a maximum decrease of the upper bound calculated for the surrogate ranking.

An expected sample complexity bound is derived for the γ-sensitive Copeland ranking procedure along with the MRD distance in a similar way like Kalyanakrishnan (2011) and Kalyanakrishnan et al. (2012). The bound is of the form $\mathcal{O}\left(R_1 \log\left(\frac{R_1}{\delta}\right)\right)$, where $R_1 = R_1(\gamma, \epsilon, \mathbf{Q})$ is a task/problem dependent constant. More specifically, R_1 depends on the (sorted) calibrated pairwise preference probabilities $\Delta_{i,j}$, and on the robustness of the ranking procedure to small changes in the preference matrix (i.e., on how much the ranking produced by the ranking procedure might be changed in terms of the MRD distance if the preference matrix is slightly altered) as well as on the desired approximation quality ϵ. An expected sample complexity is also derived for the γ-insensitive sum of expectations ranking procedure along with the MRD distance with a similar flavor as for the γ-sensitive Copeland ranking procedure. The analysis of the NDP distance is more difficult, since small changes in the preference matrix may strongly change the ranking in terms of the NDP distance. The sample complexity analysis for this distance has therefore been left as an open question.

4.2.7 Borda-Ranking

Falahatgar et al. (2017a) consider the (ϵ, δ)-PAC learning scenario with respect to the nearly optimal Borda winner resp. Borda ranking. Here, an arm a_i such that $q_i \geq \max_j q_j - \epsilon$ is called an ϵ-Borda optimal arm (cf. Section 2.2.5). A permutation $\pi \in \mathbb{S}_K$ such that $q_{\pi^{-1}(k)} \geq q_{\pi^{-1}(j)} - \epsilon$ holds for $1 \leq j < k \leq K$, is called an ϵ-Borda ranking.

The authors show that the problem of finding an ϵ-Borda optimal arm can be solved using $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\delta}\right)$ many pairwise comparisons. This is done by showing that PAC-optimal algorithms for the standard MAB setting can be used to solve the Borda score setting using the so-called *Borda reduction* of the dueling bandits to the standard MAB problem. This reduction is based on the fact that the pull of an arm a_i with an expected reward equal to the Borda score q_i can be simulated by conducting a duel of arm a_i with another randomly selected arm and returning a reward of 1 in case a_i has won and 0 otherwise (i.e., a Bernoulli MAB problem).

For the problem of finding an ϵ-Borda ranking, they present the Borda-ranking algorithm that requires $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{K}{\delta}\right)$ comparisons. The Borda-ranking algorithm first approximates the Borda score of each arm with an approximation error of $\epsilon/2$ with confidence at least $1-\delta/K$ by using the Hoeffding's inequality and the Borda reduction. Then the arms are simply sorted according to the approximated scores, which, thanks to an underlying Bonferroni correction, results in an ϵ-Borda optimal ranking with confidence at least $1-\delta$.

4.2.8 Round-Efficient Dueling Bandits

Within the scenario as presented in Section 3.2.6, Lin and Lu (2018) also analyze the goal of finding a Borda winner in an (ϵ,δ)-PAC learning scenario. To this end, they modify their original algorithm by letting all arms be active throughout the time and replacing their utility estimates by suitable Borda score estimates. The rationale behind keeping all arms active is that each active arm is dueled with a randomly chosen arm (cf. *Borda reduction* in Section 4.2.7), which allows to derive unbiased estimates of the Borda scores. Although the authors do not mention the explicit stopping criterion for this modified algorithm, the proof of its theoretical guarantees suggests that the algorithm is just run for a specific number of times $T_{\epsilon,\delta}$, which needs to be chosen appropriately, and then returns the arm with the largest estimated Borda score. By using $T_{\epsilon,\delta}=\mathcal{O}\left(\frac{1}{\epsilon^2}\log\frac{K}{\epsilon\delta}\right)$, it is shown that, with probability at least $1-\delta$, an ϵ-Borda optimal arm is returned. Consequently, the round complexity is of the same order as the chosen $T_{\epsilon,\delta}$, while the pairwise sampling complexity is K times this order. Regarding the results of Falahatgar et al. (2017a) in the previous section, it seems that the $\log(\epsilon^{-1})$ term occurring in these bounds could in fact be eliminated[16].

4.2.9 Hamming-LUCB

Heckel et al. (2018) consider the task of partitioning $\mathcal{A}$ into the top-k arms, say $\mathcal{A}_k$, and its complement $\mathcal{A}_k^{\complement}$ with respect to the Borda scores in an (ϵ,δ)-PAC setting. To this end, the approximation quality of a suggested partition $(\hat{\mathcal{A}}_k,\hat{\mathcal{A}}_k^{\complement})$ is assessed by means of the Hamming distance defined as $D_H(A,B):=|(A\cup B)\setminus(A\cap B)|$ for any subsets $A,B\subseteq\mathcal{A}$. In light of this, the approximation quality ϵ takes only values in the non-negative integers.

The authors present the Hamming-LUCB algorithm inspired by the Lower-Upper Confidence Bound (LUCB) algorithm (Kalyanakrishnan et al., 2012), which maintains a set consisting of the current $k-\epsilon$ best arms by considering the Borda score counterpart of the lower confidence bound of the $(k-\epsilon)$th best arm and a set of the current $n-k-\epsilon$ worst arms based on the Borda score upper confidence bound of the $(n-k-\epsilon)$th worst arm. The guiding principle underlying the design of Hamming-LUCB is to be confident as quickly as possible that the former set contains arms with larger Borda scores than the latter set by dueling an arm, which potentially belongs, based on its current Borda score confidence intervals, to both the actual $k-\epsilon$ best arms and the actual $n-k-\epsilon$ worst arms, with a randomly chosen arm (cf. *Borda reduction* in Section 4.2.7). Because there is

16. Note that Lin and Lu (2018) assume that the learner obtains K many pairwise preference observations per iteration, while Falahatgar et al. (2017a) assume only one. Consequently, the *pairwise* sampling complexity by Lin and Lu (2018) can be compared with the sample complexity by Falahatgar et al. (2017a).

potentially more than one arm with the latter property, the one with the widest confidence interval is chosen (possibly breaking ties). As soon as the lower confidence bound of the current $(k-\epsilon)$th best arm exceeds the upper confidence bound of the current $(n-k-\epsilon)$th worst arm, the Hamming-LUCB algorithm returns the k currently best arms as $\hat{\mathcal{A}}_k$ and correspondingly $\hat{\mathcal{A}}_k^{\complement}$ as $\mathcal{A}\backslash\hat{\mathcal{A}}_k$.

Assuming that all Borda scores are distinct, it is shown that, in order to return a partition that is (2ϵ)-close to the true underlying partition in terms of the Hamming-distance with probability at least $1-\delta$, Hamming-LUCB requires a number of comparisons of order

$$\mathcal{O}\Big(\log\Big(\frac{K}{\delta}\Big)\Big(\epsilon \cdot f_1(\Delta^{\mathrm{SE}}_{(k-\epsilon),(k+1+\epsilon)}) + \sum_{i=1}^{k-\epsilon} f_1(\Delta^{\mathrm{SE}}_{(i),(k+1+\epsilon)}) + \sum_{i=k+1+\epsilon}^{K} f_1(\Delta^{\mathrm{SE}}_{(k-\epsilon),(i)})\Big)\Big), \quad (16)$$

where $\epsilon \in \mathbb{N}_0$, $f_1(x) = \frac{\log 2\log(2/x)}{x^2}$ and for $i<j$ the difference between the Borda scores of the arms with the ith and jth best Borda score (i.e., $q_{(i)}, q_{(j)}$) is denoted by $\Delta^{\mathrm{SE}}_{(i),(j)} = q_{(i)} - q_{(j)}$. Further, they show a lower bound on the sample complexity for any algorithm, which is an (ϵ,δ)-PAC algorithm for the considered problem scenario for any preference relation $\mathbf{Q}$ satisfying $q_{i,j} \geq 3/8$ for any distinct $i,j \in [K]$, of order

$$\Omega\left(\log\Big(\frac{1}{\delta}\Big)\Big(\sum\nolimits_{i=1}^{k-2\epsilon} f_0(\Delta^{\mathrm{SE}}_{(i),(k+1+2\epsilon)}) + \sum\nolimits_{i=k+1+2\epsilon}^{K} f_0(\Delta^{\mathrm{SE}}_{(k-2\epsilon),(i)})\Big)\right), \quad f_0(x) = x^{-2}.$$

In fact, an even more refined lower bound is derived that also involves the Borda score gaps of arms having a Borda score close to the top-k arm with respect to the Borda scores. Nevertheless, the authors show that, up to logarithmic terms, Hamming-LUCB is already optimal with respect to its sample complexity, if compared with the lower bound above.

Finally, it is shown that imposing a parametric assumption on the underlying pairwise preference probabilities such as the Bradley-Terry model in (8) does not lead to qualitatively stricter lower bounds for the sample complexity.

4.2.10 Preference-based Racing

The learning problem considered by Busa-Fekete et al. (2013) is to find, for some $k<K$, the top-k arms with respect to the Copeland, Borda as well as random walk ranking procedures with high probability, while simultaneously keeping the sampling complexity as low as possible. To this end, three different learning algorithms are proposed in the finite horizon case, with the horizon given in advance. In principle, these learning problems are very similar to the value-based racing task (Maron and Moore, 1994, 1997), where the goal is to select the k arms with the highest means. However, in the preference-based case, the ranking over the arms is determined by the ranking procedure instead of the means. Accordingly, the algorithms proposed by Busa-Fekete et al. (2013) consist of a successive selection and rejection strategy. The sample complexity bounds of all algorithms are of the form $\mathcal{O}(\sum_{1\leq i<j\leq K} \Delta_{i,j}^{-2} \log K/\delta)$. Thus, they are not as tight in the number of arms as those considered in Section 3. This is mainly due to the lack of any assumptions on the structure of $\mathbf{Q}$. Since there are no regularities, and hence no redundancies in $\mathbf{Q}$ that could be exploited, a sufficiently good estimation of the entire relation is needed to guarantee a good approximation of the target ranking in the worst-case.

4.2.11 Voting Bandits

Urvoy et al. (2013) consider a general bandit learning setting involving N many (unknown) distributions $P_1, \ldots, P_N$ with respective means $\mu_1, \ldots, \mu_N \in [0,1]$, a decision set $\mathcal{D}$ as well as a known utility function $u : \mathcal{D} \times [0,1]^N \to \mathbb{R}_+$, which gives rise to the optimal decision (set) by means of $\mathrm{argmax}_{d \in \mathcal{D}} u(d, \mu)$. The goal is to find, with high probability, an optimal decision in the typical bandit learning protocol, i.e., only one distribution can be queried at a time to obtain a sample, by using as few as possible sample queries to the distributions.

By choosing these features in a suitable way, this general learning setting can be customized to the problem of finding an optimal arm in the value-based MAB or the dueling bandits learning scenario, respectively. Indeed, the dueling bandits learning scenario, for instance, can be recovered by setting the decision set as $[K]$, $N = \binom{K}{2}$ and identifying the distributions $P_1, \ldots, P_N$ with the Bernoulli distributions corresponding to the random outcomes of duels, so that $\mu_1, \ldots, \mu_N$ correspond to the entries in the upper triangle matrix of $\mathbf{Q}$. In light of this, one obtains the problem of finding a Copeland winner, by using the utility function $u(i, \mathbf{Q}) = d_i$, while the utility function $u(i, \mathbf{Q}) = q_i$ leads to the problem of finding a Borda winner[17]. Assuming the existence of a Condorcet winner, the problem of finding the latter can be modeled by using the same utility function as in the case of Copeland winner.

For the general bandit learning scenario, the authors propose the *Sensitivity Analysis of Variables for Generic Exploration* (SAVAGE) algorithm, which maintains an N-dimensional confidence region for the mean vector $(\mu_1, \ldots, \mu_N)^\top$ as well as a set $A \subset [N]$ indicating which distribution is (still) relevant for the final optimal decision. In order to make the latter set meaningful, the SAVAGE algorithm relies on a subroutine IndepTest, which needs to be customized for the concrete task of the bandit problem at hand, by exploiting structural properties of the utility function in a suitable way. The key idea is that by constantly observing samples from the underlying distributions, the confidence region will concentrate around the singleton set consisting of the mean vector and, in the course of time, allows one to infer which distribution is irrelevant for the final decision and, consequently, can be removed from the set under consideration. To this end, SAVAGE simply queries a sample from each distribution within the set A in a round-robin manner, and reduces this set by means of the subroutine IndepTest.

For the problems of finding a Copeland or Borda winner, the authors design explicit subroutines IndepTest, respectively, which roughly speaking exclude an arm as soon as there exists another arm whose pessimistic Copeland/Borda score is larger than the optimistic Copeland/Borda score of the former. Under the assumption of an existing Condorcet Winner, the authors propose to instantiate IndepTest such that it removes an arm simply by checking if this arm is even pessimistically preferred over another arm, which can be expressed as a condition on the utility function used in the Copeland case.

The sample complexity of SAVAGE for finding the best arm (Copeland or Borda winner) with probability at least $1 - \delta$ is shown to be of order

$$\mathcal{O}\left(\sum\nolimits_{1 \le i < j \le K} \frac{1}{\Delta_{i,j}^2} \log \frac{K}{\delta \Delta_{i,j}} \right),$$

17. For sake of convenience, we abuse here the notation by using $\mathbf{Q}$ in the second argument of u .

by implicitly assuming that there are no ties in $\mathbf{Q}$, so that $\Delta_{i,j} \neq 0$ for all distinct i, j.

Further, the authors suggest to use SAVAGE for the task of regret minimization if the time horizon is known in advance. To this end, they adopt an explore-then-exploit strategy in the spirit of IF or BTM (cf. Section 3.1.1 and 3.1.2). However, for all winner concepts, the resulting regret bounds are of the form

$$\mathcal{O}\left(\sum\nolimits_{1 \leq i < j \leq K} \frac{1}{\Delta_{i,j}^2} \log(KT)\right),$$

which are in general not strict for the Condorcet winner (cf. Section 3.1.7) or the Copeland winner (cf. Section 4.2.2) as the target.

4.2.12 Successive Elimination

Interested in finding the best arm according to the Borda-score with a small number of comparisons, Jamieson et al. (2015) consider a specific type of structural constraint on the preference relation $\mathbf{Q}$, which assumes a (small) set of arms, the top candidates, that are similar to each other with respect to their mutual pairwise preference probabilities, and a (large) set of arms that would be barely preferred over one randomly chosen arm among the top candidates, i.e., arms having a large Borda score gap to the top candidates. This assumption imposes some kind of sparsity on the preference relation and is motivated by numerous practical problem scenarios.

They first show that, under such a sparsity assumption, the Borda reduction (cf. Section 4.2.7) in combination with a suitable best arm identification algorithm of the value-based MAB problem may result in a poor sample complexity. Subsequently, they propose the Successive Elimination with Comparison Sparsity (SECS) algorithm, which essentially invokes the successive elimination (SE) algorithm of Even-Dar et al. (2006) for the value-based MAB problem with the Borda reduction, but enhances upon the latter straightforward approach by explicitly making use of the possible sparsity of the underlying preference relation $\mathbf{Q}$. To this end, partial Borda score gaps are considered, where the Borda score in (13) is computed only on a subset of $\mathcal{A}$. More specifically, for a specific number of iterations T_0 (input of SECS), the SECS algorithm uses only the elimination criterion of SE, which removes arms having large (estimated) Borda score gaps to any another arm. After exceeding T_0 iterations, the extra elimination criterion is also activated, which eliminates arms having a large partial Borda score gap estimated by using a set of potential top candidates, thereby exploiting a possible sparsity.

Under the above sparsity assumption and uniqueness of the Borda winner, a high probability upper bound on the sample complexity of SECS of order

$$\mathcal{O}\Big(\sum_{i \neq i^*} \min\Big\{ \max\Big\{ \frac{1}{\Delta_{min}^{\mathrm{SE}}} \log \frac{K}{\Delta_{min}^{\mathrm{SE}}\delta}, \frac{1}{K(\Delta_i^{\mathrm{SE}})^2} \log \frac{K}{(\Delta_i^{\mathrm{SE}})^2\delta} \Big\}, \frac{1}{(\Delta_i^{\mathrm{SE}})^2} \log \frac{K}{(\Delta_i^{\mathrm{SE}})^2\delta} \Big\}\Big) \tag{17}$$

is shown, where i^* denotes the index of the arm with the largest Borda score, $\Delta_{min}^{\mathrm{SE}} = \min_{i \neq i^*} \Delta_i^{\mathrm{SE}}$ and $\Delta_i^{\mathrm{SE}} = q_{i^*} - q_i$. Here, the first term within the minimum is representing the improvement over the straightforward SE with Borda reduction, whose sample complexity would correspond to the second term within the minimum. Further, a lower bound of order $\Omega\left(\sum_{i\neq i^*} \frac{1}{(q_{i^*}-q_i)^2} \log \frac{1}{\delta}\right)$ on the expected sample complexity of finding the Borda winner with confidence $1-\delta \in (0.85, 1)$ is shown, which holds if $q_{i,j} \in [3/8, 5/8]$ for any pair $i, j \in [K]$.

4.2.13 Bayesian Sequential Sampling

Seeking to find the top-k Borda scored arms as in (Heckel et al., 2018), the work by (Groves and Branke, 2019) investigates adaptations of well-established Bayesian methods used in the Simulation Optimization community (Branke et al., 2007) to obtain learning algorithms with a small sample complexity. For this purpose, they consider the *probability of correct selection* given by

$$\mathbf{P}(q_i > q_j \, \forall i \in \mathcal{S},\, j \notin \mathcal{S}), \quad \mathcal{S} \subset [K], |\mathcal{S}| = k,$$

as the performance metric for the learner. This metric is estimated by the *approximated expected probability of correct selection* (AEPCS) defined by

$$\mathbf{P}\Big(\big(\bigcap\nolimits_{i\in\hat{\mathcal{S}}} \{q_i > c\}\big) \cap \big(\bigcap\nolimits_{j\notin\hat{\mathcal{S}}} \{q_j < c\}\big) \Big), \quad c > 0,$$

where $\hat{\mathcal{S}} \subset [K]$ is the k-sized set of arms with the current highest estimated Borda scores. The consideration of AEPCS is reasonable as it is a lower bound for the sample version of the probability of correct selection, i.e., $\mathbf{P}(q_i > q_j \, \forall i \in \hat{\mathcal{S}},\, j \notin \hat{\mathcal{S}})$. Based on a normal distribution assumption on the estimated Borda scores, the Pairwise Optimal Computing Budget Allocation (POCBA) algorithm is suggested, which samples in each round the pair of arms having the maximal expected increase on the AEPCS until the AEPCS criterion reaches a predefined threshold.

The authors further suggest to approximate the expected value of information gain of a single duel by the probability that the outcome of the duel will change the current set consisting of the highest estimated Borda scores $\hat{\mathcal{S}}$. Exploiting again the normal assumption on the estimated Borda scores, the authors propose the Pairwise Knowledge Gradient (PKG) algorithm, which is successively choosing the pair of arms maximizing the approximated information gain based on its current Borda score estimates. The algorithm terminates similar as POCBA as soon as the AEPCS criterion exceeds some specific threshold.

Both algorithms are shown to be asymptotically optimal, that is, if the algorithms are run with an infinite sampling budget the probability of a correct selection is tending to one.

4.2.14 Active Ranking

The coarse ranking problem is the task of sorting random variables according to a specific parameter of their associated distribution into clusters of predefined sizes (Katariya et al., 2018)[18]. Concretely, for a given number of clusters $c \in \{2, 3, \dots, K\}$ and for given cluster boundaries $1 \le k_1 < k_2 < \ldots < k_c = K$ it is intended to identify the k_1 random variables with the "best" parameters, and from the remaining $K - k_1$ ones the $k_2 - k_1$ random variables with the "best" parameters, and so forth. Transferring this idea to the realm of

18. The minimal requirement on these parameters is that they admit an order relation.

bandit problems, it is quite natural to use the expected value of an arm's reward distribution as the parameter of interest in the value-based MAB problem. In the dueling bandits setting, on the other side, the choice of a suitable parameter is again far from obvious, due to the absence of numerical rewards. Heckel et al. (2019) propose to use the Borda score of an arm as the parameter of interest and study the sample complexity of algorithms to identify a coarse ranking with high probability. It is worth noting that the coarse ranking task is a generalization of finding the best arm ($k_1 = 1, k_2 = K$) and identifying the top-k arms problem ($k_1 = k, k_2 = K$), and also includes the task of finding a total ranking over the arms ($k_i = i$ for $i \in [K]$).

The authors propose the Active Ranking (AR) algorithm, which uses the idea underlying racing or successive elimination strategies (Paulson, 1964; Maron and Moore, 1994; Even-Dar et al., 2006). More specifically, AR maintains a set of active arms consisting of arms which have not been assigned to a cluster yet, from which one arm is dueled with another arm chosen uniformly at random from the set of all arms (without the first arm) in a round-robin manner. Once the upper and lower confidence bounds on the Borda scores of the active arms reveal that one of the active arms belongs to a certain cluster, the latter is assigned to this cluster and removed from the set of active arms.

Assuming that all Borda scores are distinct, the authors show that AR outputs the correct coarse ranking with probability at least $1-\delta \in [0.86, 1)$ and has a sample complexity of the order

$$\mathcal{O}\Big(\log(K/\delta)\Big(\sum_{i=1}^{k_1} f_1(\Delta^{\mathrm{SE}}_{(i),(k_1+1)}) + \sum_{l=2}^{c-1} \sum_{i=k_{l-1}+1}^{k_l} \max\big\{ f_1(\Delta^{\mathrm{SE}}_{(k_{l-1}),(i)})\,,\, f_1(\Delta^{\mathrm{SE}}_{(i),(k_l+1)})\big\} \\ + \sum_{i=k_{c-1}+1}^{K} f_1(\Delta^{\mathrm{SE}}_{(k_{c-1}),(i)})\Big)\Big) \;, \tag{18}$$

where $f_1(x) = \frac{\log 2 \log(2/x)}{x^2}$ and $\Delta^{\mathrm{SE}}_{(i),(j)}$ is as in Section 4.2.9. Additionally, a lower bound for any algorithm that outputs the correct coarse ranking with probability at least $1-\delta \in [0.86, 1)$ is shown:

$$\Omega\Big(\log(1/\delta)\Big(\sum_{i=1}^{k_1} f_0(\Delta^{\mathrm{SE}}_{(i),(k_1+1)}) + \sum_{l=2}^{c-1} \sum_{i=k_{l-1}+1}^{k_l} \max\big\{ f_0(\Delta^{\mathrm{SE}}_{(k_{l-1}),(i)})\,,\, f_0(\Delta^{\mathrm{SE}}_{(i),(k_l+1)})\big\} \\ + \sum_{i=k_{c-1}+1}^{K} f_0(\Delta^{\mathrm{SE}}_{(k_{c-1}),(i)})\Big)\Big) \;,$$

where $f_0(x) = x^{-2}$. This lower bound holds if the underlying preference relations $\mathbf{Q}$ is such that $q_{i,j} \geq 3/8$ for any distinct $i, j \in [K]$ and shows that AR is nearly optimal.

Last but not least, similar as in the PAC learning scenario in Section 4.2.9, it is shown that parametric assumptions on the underlying pairwise preference probabilities do not lead to qualitatively stricter lower bounds for the sample complexity for the considered coarse ranking task.

5. Further Extensions

In this section, we review different generalizations and extensions of the setting of preference-based (dueling) bandits as discussed in the previous sections.

5.1 Adversarial Dueling Bandits

Ailon et al. (2014b) consider the adversarial variant of the utility-based dueling bandits problem, in which no stochastic assumption on the utilities of the arms is required, i.e., the latent utility of each arm may change over iterations. The authors suggest to apply the reduction algorithm SPARRING, which has originally been designed for stochastic settings, with an adversarial bandit algorithm such as the Exponential-weight algorithm for Exploration and Exploitation (EXP3) (Auer et al., 2002b) as a black-box MAB. More specifically, the algorithm uses two separate MABs, one for each arm. On receiving a relative feedback about a duel, one instantiation of EXP3 only updates its weight for one arm and the other instantiation only updates for the other arm. For this SPARRING reduction, it is shown that the $\mathcal{O}(\sqrt{KT \ln K})$ upper bound of EXP3 in the adversarial MAB problem is preserved.

Adversarial utility-based dueling bandits are also studied by Gajane et al. (2015). They suggest the Relative Exponential-weight for Exploration and Exploitation (REX3) algorithm, which is an extension of the EXP3 algorithm to the dueling bandits setting with feedback in the form of pairwise preferences. The key property observed by the authors for the optimal arm in expectation in a particular time step is that, in addition to its property of maximizing the absolute reward, it is also the one maximizing the regret of any fixed opponent strategy (a role that might be played by the algorithms' strategy itself). This observation provides a possibility to estimate the individual rewards of two arms involved in a comparison, despite having access only to a relative value, and thus allowing them to adapt the EXP3 algorithm to the dueling bandits setting. In addition to providing a general lower bound of order $\Omega(\sqrt{KT})$ on the regret of any algorithm, using the reduction to the classical MAB problem by Ailon et al. (2014b), they prove an upper bound on the expected regret of order $\mathcal{O}(\sqrt{KT \ln K})$ for REX3, which is of the same order as the regret bound of EXP3 for adversarial MABs, and the one by Ailon et al. (2014b).

Finally, Zimmert and Seldin (2019) consider the adversarial utility-based dueling setting as well and feed the SPARRING algorithm with their proposed TSALLIS-INF algorithm for the adversarial MAB problem. This leads to an $\mathcal{O}(\sqrt{KT \ln K})$ upper bound of SPARRING. In addition, they consider the stochastically constrained adversarial setting, in which, contrary to the adversarial setting, a best arm is fixed throughout the time horizon. For this setup, they show an expected regret upper bound for TSALLIS-INF of order $\mathcal{O}(\sum_{j \neq i^*} \frac{\ln(T)}{\Delta_{i^*,j}}) + \mathcal{O}(K)$ resulting in an expected regret upper bound of the same order for SPARRING in the stochastically constrained adversarial version of the utility-based dueling bandits setting.

5.2 Contextual Dueling Bandits

Dudík et al. (2015) extend the dueling bandits framework to incorporate contextual information in the spirit of MAB problems (Auer, 2002). More precisely, the learner is supposed to optimize its choice of arms in the course of an iterative learning process of the following kind: In each round, the learner observes a random context, chooses a pair of actions, conducts a duel between them, and observes an outcome in the form of a pairwise preference. The authors consider the solution concept of a von Neumann winner and present three algorithms for online learning and for approximating such a winner from batch-like data, while measuring performance of a learner by means of the regret as in Section 4.2.4.

The authors first present an algorithm that shares similarities with the sparring approach of Ailon et al. (2014b) discussed in Section 5.1: Two separate independent copies of the multi-armed bandit algorithm Exp4.P (Beygelzimer et al., 2011), which is designed to work with a pre-specified space of multi-armed bandit algorithms in a contextual setting, are run against each other. Both copies are using the same action space, context space and space of multi-armed bandit algorithms as for the original problem. In each round, the environment (or nature) specifies a context and a preference matrix, while only the context is revealed to both copies, which then select an action, respectively. A duel is run between the two selected actions, and the (negated) outcome (cf. the zero-sum game matrix in Section 4.1) is forwarded as feedback to the first (second) copy. This approach, called Sparring Exp4.P, leads to a regret that is upper bounded by $\mathcal{O}(\sqrt{KT\ln(|\Pi|/\delta)})$, where Π is the space of MAB algorithms, with probability at least $1-\delta$, and requires time and space linear in the term $|\Pi|$, which leads to computational issues in cases where Π is large.

To deal with the latter issue, the authors furthermore propose a general approach for constructing an approximate von Neumann winner. To this end, the problem of finding a von Neumann winner is substituted by a more convenient empirical version of the latter problem. Assuming the existence of a classification oracle on Π, which can find the minimum cost MAB algorithm in Π if it is provided with the cost of each action on each sequence of contexts, the authors propose two algorithms: SparringFPL, which is sparring two copies of the Follow-the-Perturbed-Leader algorithm (Kalai and Vempala, 2005), and ProjectedGD, which builds on the projected gradient descent algorithm (Zinkevich, 2003), but is sparring essentially against its worst-case opponent strategy. The two algorithms are primarily solving a compact game based on the more convenient empirical problem version, which, however, is equivalent to computing an approximate von Neumann winner. While their regret bound is weaker than for Sparring Exp4.P, namely $\mathcal{O}((KT)^{2/3}\ln^{1/3}(|\Pi|/\delta))$, they require time and space that depend only logarithmically on $|\Pi|$. Further, the running time as well as the number of oracle calls in order to return an approximate von Neumann winner is theoretically analyzed.

Based on the Perceptron algorithm, Cohen and Crammer (2014) develop the SHAMPO (SHared Annotator for Multiple PrOblems) algorithm for online multi-task learning with a shared annotator, in which learning is performed in rounds. In each round, each of K different learners receives an input and predicts its label. A shared stochastic mechanism then annotates one of the K inputs, and the learner receiving the feedback updates its prediction rule. The authors show that this algorithm can be used to solve the contextual dueling bandits problem when a decoupling of exploration and exploitation is allowed.

To pick a task to be labeled, SHAMPO performs an exploration-exploitation strategy in which tasks are randomly queried, with a bias towards tasks that involve a high uncertainty about the labeling. To perform an update on the parameter vector representing the model, the algorithm applies the Perceptron update rule to the true label revealed for the task chosen.

5.3 Dueling Bandits on Posets

Audiffren and Ralaivola (2017) extend the dueling bandits problem to partially ordered sets (posets), allowing pairs of arms to be incomparable. They consider the problem of identifying, with a minimal number of pairwise comparisons in a pure exploration setting, the set of maximal arms or Pareto set among all available arms. The main challenge in this framework is the problem of indistinguishability: The learner may be unsure whether two arms are actually comparable and just very close to each other, or whether they are indeed incomparable, no matter how many times the arms are compared. Without any additional information, it might then be impossible to recover the exact Pareto set.

The authors first devise the UnchainedBandits algorithm to find a nearly optimal approximation of the Pareto front of any poset. The strategy implemented by the algorithm is based on a peeling approach that offers a way to control the number of comparison of arms that are in fact indistinguishable. The authors provide a high probability regret bound of $\mathcal{O}\left(K \,\mathrm{width}(S) \log \frac{K}{\delta} \sum_{i, i \notin \mathcal{P}} \frac{1}{\Delta_i}\right)$, where S is the poset, $\mathrm{width}(S)$ is its width defined as the maximum size of an antichain (a subset in which every pair is incomparable), $\mathcal{P}$ is the Pareto front, Δ_i is the regret associated with arm a_i defined as the maximum difference between arm a_i and the best arm comparable to a_i, and the regret incurred by comparing two arms a_i and a_j is defined by $\Delta_i + \Delta_j$.

Further, by making use of the concept of decoys, the authors show that UnchainedBandits can recover the exact set $\mathcal{P}$, incurring regret that is comparable to the former one—except for an extra term due to the regret incurred by the use of decoys—with a sample complexity that is upper bounded by $\mathcal{O}(K \mathrm{width}(S) \log(NK^2/\delta)/\Delta^2)$, where N is a positive integer related to a weaker form of distinguishability and Δ is a parameter of the decoys. The concept of decoys is an idea inspired by works from social sciences and psychology, intended to force a decision maker to choose a specific option (the target option) by presenting her/him a choice between the target option and another option dominated by the latter in specific aspects (the decoy option).

5.4 Graphical Dueling Bandits

Di Castro et al. (2011) consider the bandit problem over a graph, the structure of which defines the set of possible comparisons. More specifically, they assume that there is an inherent and unknown value per node (arm), and that the graph describes the allowed comparisons: two nodes are connected by an edge if they can be dueled with each other. Such a duel returns a random number in $[-1, 1]$, the expected value of which is the difference between the values of the two nodes. Thus, unlike the traditional dueling bandits setup, the topology is not a complete graph, and non-adjacent nodes can only be compared indirectly by sampling all the edges along a path connecting them. Further, the learner receives feedback in the form of a numerical value.

The authors consider different topologies and focus on the sample complexity for finding the optimal arm (largest latent value) in the PAC setting. For the linear topology, in which each node is comparable to at most two other nodes, they present an algorithm that samples all edges, computes the empirical mean of each edge, and based on these statistics, assigns a value to each node reflecting its alleged optimality. The sample complexity is $\mathcal{O}(\frac{K^2}{\max\{\epsilon,u\}^2}\log(\frac{1}{\delta}))$, where u is the difference between the node with the highest value and the node with the second highest value.

For the tree topology, that is, a topology in the form of a tree, the authors present an algorithm which essentially reduces the graphical bandit problem to graphical bandit problems with linear topologies by treating each path from the root to a leaf as a line graph and use one each of these graphs the algorithm above. This leads to a sample complexity of $\mathcal{O}(\frac{KD}{\max\{\epsilon,u\}^2}\log(\frac{|L|}{\delta}))$, where D is the diameter of the tree and L the set of leaves.

For the network topology, that is, general connected and undirected graphs, the authors present the Network Node Elimination (NNE) algorithm, which is inspired by the action elimination procedure of Even-Dar et al. (2006). This algorithm has a sample complexity upper bounded by $\frac{KD}{(\max\{\epsilon,u\}/\log K)^2}\log(\frac{K}{\delta/\log K})$. Further, they consider the contextualized version of the problem in the spirit of Section 5.2, and show that a version of the NNE algorithm achieves a sample complexity of the form $\mathcal{O}(B\log^2 B)$, where $B = \frac{KD}{(\epsilon/\log K)^2}\log\left(\frac{K}{\delta/\log K}\right)d^2$, and d is the dimension of the feature vectors.

Karnin (2016) also applies the suggested verification algorithm to graphical bandits, which improves upon the latter approach by logarithmic terms.

5.5 Dueling Bandits with Dependent Arms

Focusing once again on minimizing the weak regret, Chen and Frazier (2016) study the utility-based dueling bandits, where each utility $u(\theta, x_i)$ of an arm a_i is determined by a known utility function $u : \mathbb{R}^{d'} \times \mathbb{R}^d \to \mathbb{R}$ of a known (and fixed) d-dimensional arm-specific feature x_i and some unknown d'-dimensional weight parameter θ. Thus, dependency among arms can be modeled if d' is smaller than d, which in turn can be exploited by a learner.

The authors introduce the Comparing The Best (CTB) algorithm, which maintains 2^K many cells, one for each possible pairwise preference sequence involving every pair of arms, and assigns scores to each cell based on the number of times a pairwise preference corresponding to the cell has been observed. In each iteration, the first arm of the duel is the one that is optimal with respect to the cell(s) with the largest score, while the second arm is the arm that is optimal with respect to the cell(s) having largest score, and where the optimal arm is different from the first one.

It is shown that CTB enjoys a bound on its expected cumulative weak regret of order $\mathcal{O}\left(\frac{K^2 M'}{\min_{i\neq j}\Delta_{i,j}^2}(\max_{i,j} u(\theta,x_i) - u(\theta,x_j))\right)$, where M' is a constant which is $\Theta(2^K)$ in general, but $O(K^{2d'})$ for special cases.

Further, under specific conditions, the authors show that the algorithm can be implemented in a favorable way in order to cope with high computational costs due to the large number of cells.

5.6 Partial Monitoring Games

The dueling bandits problem can be seen as a special case of the partial monitoring (PM) problem (Bartók et al., 2011; Bartók, 2013; Bartók et al., 2014)—a generic model for sequential decision-making with incomplete feedback, which is defined by a quintuple $(\mathbf{N}, \mathbf{M}, \mathbf{\Sigma}, \mathcal{L}, \mathcal{H})$, where $\mathbf{N}$ is the set of actions, $\mathbf{M}$ is the set of outcomes, and $\mathbf{\Sigma}$ is the feedback alphabet; the loss function $\mathcal{L}$ associates a real-valued loss $\mathcal{L}(I, J)$ with each action $I \in \mathbf{N}$ and outcome $J \in \mathbf{M}$, and the feedback function $\mathcal{H}$ associates a feedback symbol $\mathcal{H}(I, J) \in \mathbf{\Sigma}$. In each round of a PM game, first the opponent chooses an outcome J_t from $\mathbf{M}$, and the learner an action I_t from $\mathbf{N}$. Then, the learner suffers the loss $\mathcal{L}(I_t, J_t)$ and receives the feedback $\mathcal{H}(I_t, J_t)$. The performance of a learner is measured by means of the expected cumulative regret against the best single-action strategy

$$R^T = \max_{i \in \mathbf{N}} \sum_{t=1}^{T} \mathcal{L}(I_t, J_t) - \mathcal{L}(i, J_t). \tag{19}$$

Following Gajane and Urvoy (2015) the utility-based dueling bandits problem with a linear link function (and ties) can be encoded as a PM problem with the set of actions given by the set of all pairs of arms $\mathbf{N} = \{(i, j) : 1 \le i, j \le K, i \le j\}$, the alphabet $\mathbf{\Sigma} = \{0, 1/2, 1\}$, and the set of outcomes given as vectors $\mathbf{m} = (m_1, \ldots, m_K) \in \mathbf{M} = [0, 1]^K$, where m_i is simply the utility of arm a_i, which can be interpreted as its instantaneous gain, so that we differ from the notation as in Section 3.2. After the environment selects an outcome $\mathbf{m} \in \mathbf{M}$ and the learner a duel $(i, j) \in \mathbf{N}$, the instantaneous loss[19] is

$$\mathcal{L}((i, j), \mathbf{m}) = 1 - \frac{\mathbf{m}_i + \mathbf{m}_j}{2} \quad ,$$

and the feedback is

$$X = \begin{cases} 0, & \text{if } \mathbf{m}_i < \mathbf{m}_j \\ 1/2, & \text{if } \mathbf{m}_i = \mathbf{m}_j \\ 1, & \text{if } \mathbf{m}_i > \mathbf{m}_j \end{cases} .$$

Using the PM formalism, the authors prove that the dueling bandits problem is an easy instance according to the hierarchy of PM problems (Bartók et al., 2011; Bartók, 2013), i.e., it is possible to achieve a regret bound of order $\tilde{\mathcal{O}}(\sqrt{T})$ and the difference of the loss vectors of adjacent actions are locally observable, where regret is measured as in (19).

Quite recently, Kirschner et al. (2020) suggested the Information Directed Sampling (IDS) algorithm for linear partial monitoring games and also consider the utility-based dueling bandits setting above as a special case. Their results lead to a regret bound of order $\tilde{\mathcal{O}}(\sqrt{KT})$, which is the same as REX3 (cf. Section 5.1).

5.7 Dueling Bandits for Qualitative Feedback

Xu et al. (2019) consider the qualitative dueling bandits problem, which is a variant of the MAB problem with reward feedback of the pulled arm on an ordinal scale, i.e., qualitative feedback, instead on a numerical scale, i.e., quantitative feedback. This setting was introduced by Szörényi et al. (2015b), but analyzed there in a rather classical MAB setting. Instead, following ideas by Busa-Fekete et al. (2013), the authors embed the problem into a dueling bandits setting. More precisely, for each pair of arms (i, j), they define the pairwise

19. Note that Gajane and Urvoy (2015) work with the instantaneous gain $\mathcal{G}((i, j), \mathbf{m}) = \frac{\mathbf{m}_i + \mathbf{m}_j}{2}$.

winning probability in (1) as $q_{i,j} = \mathbf{P}(X_i > X_j) + 1/2\mathbf{P}(X_i = X_j)$, where X_i, X_j are mutually independent random variables with values in the considered ordinal scale. The law of X_i is ν_i, representing the qualitative feedback mechanism of an arm a_i.

With this definition of pairwise probabilities, one can orchestrate the dueling bandits mechanism for this framework by simultaneously pulling two arms and observing which one has a higher ordinal reward, breaking ties at random. The authors seek to minimize the expected cumulative regret, with the cumulative regret given as

$$\mathbf{E}[R^T] = \sum_{t=1}^{T} \Delta_{a_{i^*}, a_{i(t)}} ,$$

where a_{i^*} is either the Condorcet winner or the Borda winner, and $i(t)$ the index of the arm pulled in time step t.

For the case of the Condorcet winner, they suggest the Thompson Condorcet sampling algorithm, which uses for each arm a_i a Dirichlet distribution as the prior distribution for the probability vector specified by the law ν_i. In each time step, the algorithm generates a random sample from the posterior distribution and pulls the Condorcet winner based on these random samples[20].

For the case of the Borda winner, the Thompson Borda sampling algorithm and the Borda-UCB algorithm are introduced. The Thompson Borda sampling algorithm follows the same idea as Thompson Condorcet sampling, only replacing the determination of the Condorcet winner by the Borda winner of the posterior samples. The Borda-UCB algorithm is based on the UCB algorithm. It pulls the arm with the highest upper confidence bound on the estimated Borda score if this is also the arm with the highest number of pulls. Otherwise, all arms not having the highest number of pulls are pulled in order to enhance the estimation of the Borda scores.

Furthermore, in the case of an existing Condorcet winner, the authors show an expected regret bound for Thompson Condorcet sampling of order $\mathcal{O}(K \log(T))$. They also show that the constant appearing in the bound can be arbitrary small compared to the constants in the lower bound for the dueling bandits problem applied to the qualitative dueling bandits problem. For the case of the Borda winner, they show that Thompson Borda sampling suffers an expected regret which is polynomial with respect to T, while Borda-UCB achieves expected regret bounds of order $\mathcal{O}(K \log(T))$, which, as verified by the authors, matches the theoretical lower bound.

5.8 Combinatorial Pure Exploration for Dueling Bandits

The combinatorial pure exploration for dueling bandits problem is a variation of the combinatorial pure exploration MAB problem (Chen et al., 2014) introduced by Chen et al. (2020). In this setting, there exists a known bipartite graph $G = (\mathcal{A}, \mathcal{P}, E)$ with partitions $\mathcal{A} = \{a_1, \ldots, a_K\}$ and $\mathcal{P} = \{p_1, \ldots, p_l\}$, where $l \leq K$, and a set of edges $E \subset \mathcal{A} \times \mathcal{P}$ with the following meaning: The arms correspond to candidates, while elements in $\mathcal{P}$ are specific positions, and each edge $e_{i,k} = (a_i, p_k) \in E$ represents that arm a_i is available for position p_k. Moreover, the existence of an unknown preference matrix $\mathbf{Q}_G \in [0,1]^{|E| \times |E|}$ associated

20. If no Condorcet winner exists for the sample, a new random sample is drawn.

with G is assumed. The entries of this matrix are $(q_{e_{i,k},e_{i',k'}})_{e_{i,k},e_{i',k'}\in E}$, which are zero if $k \neq k'$ and otherwise indicate the stochastic preference of arm a_i over $a_{i'}$ for position p_k.

The goal is to find an optimal maximum matching[21] M in G based on $\mathbf{Q}_G$, where it is once again not obvious how to define the notion of an optimal maximum matching without real-valued rewards (cf. Section 2.2). As a remedy, the authors introduce the pairwise preference of a maximum matching M over a maximum matching M' (both with respect to G) by means of

$$q^G_{M,M'} = \frac{1}{l}\sum_{k=1}^{l} q_{M(k),M'(k)},$$

where $M(k)$ (or $M'(k)$) denotes the edge in M (or M') with p_k as its endpoint. With this, it is possible to leverage the winner concepts of the dueling bandits: Let $\mathcal{M}$ be the set of all maximum matchings in G, then

- M^* is a Condorcet matching winner if $q^G_{M^*,M} > 1/2$ holds for any $M \in \mathcal{M}\backslash\{M^*\}$;
- M^*_{SE} is a Borda matching winner if it has the highest Borda matching score defined for any $M \in \mathcal{M}$ by $q^G_M = \frac{1}{|\mathcal{M}|}\sum_{M'\in\mathcal{M}} q^G_{M,M'}$.

Once again, the Condorcet matching winner might not exist for $\mathbf{Q}_G$, while a Borda matching winner always exists, but might not be unique and does not necessarily coincide with the former if it exists.

For the Borda matching winner as the underlying goal, one could use the Borda reduction technique (cf. Section 4.2.7) and make use of the CLUCB for the combinatorial pure exploration MAB problem suggested by Chen et al. (2014). However, the naïve usage of the Borda reduction would require an exact uniformly at random sampling from the set of all maximum matchings in G, which in turn could lead to exponential costs. Therefore, the authors modify this approach leading to the CLUCB-Borda-PAC and the CLUCB-Borda-Exact algorithm, which replace the exact uniformly at random sampling procedure by means of a substitutional sampling procedure generating approximately uniform random samples of $\mathcal{M}$, but with polynomial costs. As the approximate sampling procedure introduces a bias into the natural estimates, both algorithms are adjusted accordingly.

CLUCB-Borda-PAC is shown to be an (ϵ,δ)-PAC algorithm for finding the Borda matching winner with a sample complexity of order $\mathcal{O}\left(\frac{1}{H^{\mathrm{SE}}_\epsilon}\log\left(\frac{H^{\mathrm{SE}}_\epsilon}{\delta}\right)\right)$, where $H^{\mathrm{SE}}_\epsilon = \sum_{e\in E}\min\{\frac{C_G^2}{(\Delta^{\mathrm{SE}}_e)^2}, \frac{1}{\epsilon^2}\}$ with Δ^{SE}_e being some specific gap terms and $C_G > 0$ some constant depending on the structure of the bipartite graph G. For CLUCB-Borda-PAC, the authors verify its correctness to return the Borda matching winner with probability at least $1-\delta$ and derive a sample complexity bound of order

$$\mathcal{O}\left(C_G^2 \cdot H^{\mathrm{SE}} \cdot \log(l/\Delta^{\mathrm{SE}}_{\min}) \cdot \log\left(\frac{C_G\, H^{\mathrm{SE}}}{\delta}\right) + \log\log(l/\Delta^{\mathrm{SE}}_{\min})\right),$$

21. A matching of a bipartite graph $G = (\mathcal{A},\mathcal{P},E)$ is a sub-graph G' with a set of edges $E' \subset E$ such that there exist no two edges $e' = (a',p'), e'' = (a'',p'')$ in E' with $p' = p''$. A maximum matching is a matching of G with the maximal possible number of edges.

where $H^{\mathrm{SE}} = \sum_{e \in E} \frac{1}{(\Delta_e^{\mathrm{SE}})^2}$ and $\Delta_{\min}^{\mathrm{SE}} = \min_{e \in E} \Delta_e^{\mathrm{SE}}$. Further, a lower bound of order $\Omega\left(H^{\mathrm{SE}} \log(1/\delta)\right)$ for this learning scenario is shown on a specific subclass of problem instances, so that both algorithms are almost optimal regarding their sample complexity for such problem instances.

Assuming the existence of a Condorcet matching winner, the authors suggest two algorithms, CAR-Cond and CAR-Parallel, for identifying it. CAR-Parallel uses k many parallel variants of the CAR-Cond algorithm, which are adapting the verification idea of Karnin (2016) (cf. Section 3.1.17), with suitably chosen confidence levels in order to profit from the variant CAR-Cond having the most favorable sample complexity of all parallel variants. The underlying key idea of CAR-Cond is that the Condorcet matching winner can be expressed as the solution of a specific optimization problem, which in turn can be relaxed to a convex optimization problem. Having access to some oracle that can return an approximate solution under some specific constraints and a suitable guarantee, the CAR-Cond algorithm maintains a set of undecided edges and uses upper as well as lower confidence estimates of the entries in $\mathbf{Q}_G$ to iteratively check whether an undecided edge e is likely an element of the Condorcet matching winner or not.

Both algorithms admit a polynomial running time and the authors derive respective bounds on the sample complexity: CAR-Cond has a bound of order

$$\mathcal{O}\left(\sum\nolimits_{j=1}^{l} \sum\nolimits_{e,e' \in E: e \neq e'} \frac{1}{(\Delta_{e,e'})^2} \log\left(\frac{K^G}{\delta \Delta_{e,e'}}\right)\right),$$

where K^G is the number of all possible duels within G, and $\Delta_{e,e'}$ is some gap term, while the sample complexity bound of CAR-Parallel is qualitatively similar to (4) by replacing K by K^G, $\Delta_{i,j}$ by $\Delta_{e,e'}$, and the sums by sums as in the display above.

6. Multi-Dueling Bandits

Although there are various practical scenarios in which the underlying sequential decision process can be modeled by means of the dueling bandits setting, because each of the decisions corresponds to a qualitative comparison of two choice alternatives, this modeling approach is apparently not sufficient to capture more general sequential decision processes in which qualitative comparisons of more than two available choice alternatives can be carried out at once. Such more general variants occur in many fields of application such as recommender systems, where a set of items (videos, songs, etc.) are displayed to a user, whereupon the latter expresses her preference over the displayed items in the form of a discrete choice. Another relevant field is web search, where usually an ordered list of the allegedly most relevant websites related to a user's query is returned, resulting in observing a click (or no-click) of the user for the presented website collection. Last but not least, any application involving social choices fits into the more general variant as well, because the underlying data correspond to individual opinions oftentimes expressed in the form of a (partial) ranking over specific choice alternatives.

Motivated by the limited coverage of the dueling bandits modeling approach for such practically relevant problem scenarios, research interest in the so-called *multi-dueling bandits* problem has recently increased. The latter is a generalization of the dueling bandits problem allowing the learner a great deal of latitude with regard to the available actions. This generalization is conceptually similar to how the *Combinatorial Bandits* (Cesa-Bianchi and Lugosi, 2012) generalize the classical value-based MAB problem by allowing the learner to select subsets of arms in each iteration as well, whereupon feedback either in the form of rewards of each single arm in the selected subset (semi-bandit feedback) or the total sum of the rewards (bandit feedback) is observed. However, just like the underlying basic variants, these two generalizations are still fundamentally different, since the feedback obtained by the learner in the multi-dueling bandits setting is of a qualitative nature, while the feedback in the combinatorial bandits is of a quantitative or numerical nature.

There are some active fields of research that are closely related to the multi-dueling bandits problem, in the sense that the learner also receives some kind of preference feedback related to the choice alternatives (arms). These frameworks and their similarities as well as differences will be discussed at the end of this section.

Due to the different and in particular more general action space compared to the basic dueling bandits problem, there are a couple of novelties emerging in the modeling of the learning scenario of multi-dueling bandit problems, which shall be described in the following.

6.1 Learning Protocol

As already pointed out, one of the main differences between the dueling and the multi-dueling bandits problem concerns the action space of the learner, which we shall denote by $\mathbb{A}$ throughout this section. In the setting of multi-dueling bandits, the action space $\mathbb{A}$ corresponds to a family of subsets of $\mathcal{A} = \{a_1, \ldots, a_K\}$, which does not necessarily have to be the family of all possible pairs of arms in $\mathcal{A}$, while the decision making process still iterates in discrete steps, either through a finite time horizon $\mathbb{T} \coloneqq [T] = \{1, \ldots, T\}$ or an infinite horizon $\mathbb{T} \coloneqq \mathbb{N}$. In each iteration $t \in \mathbb{T}$, the learner can perform as its action a comparison of the arms within the selected subset $A_t \in \mathbb{A}$, resulting in a qualitative feedback (cf. Section 6.2). Practically motivated forms of the underlying action space $\mathbb{A}$ include the following:

- $\mathbb{A}_l$, all subsets of $\mathcal{A}$ of a fixed size $l \in \{2, \ldots, K\}$.
- $\mathbb{A}_l^+$, the union of $\mathbb{A}_l$ and $\{a_1\}, \ldots, \{a_K\}$.
- $\mathbb{A}_{\leq l}$, all subsets of $\mathcal{A}$ with cardinality at least two but at most l for some fixed $l \in \{2, \ldots, K\}$.
- $\mathbb{A}_{\leq l}^+$, the union of $\mathbb{A}_{\leq l}$ and $\{a_1\}, \ldots, \{a_K\}$.
- $\mathbb{A}_{\text{full}} \coloneqq \mathbb{A}_{\leq K}^+$, all non-empty subsets of $\mathcal{A}$.

By characterizing the admissible "full commitment" to one arm of the dueling bandits by means of the singleton sets $\{a_1\}, \ldots, \{a_K\}$, the action space of the basic dueling bandits setting corresponds to the action space $\mathbb{A}_{\leq 2}^+$. Needless to say, the complexity of most of the above action spaces is much higher than the complexity of the action space underlying the dueling bandits.

6.2 Feedback Mechanisms

The fundamental assumption of observing a qualitative feedback underlying the dueling bandits or basic preference-based multi-armed bandits can be manifested in various ways if the learner expects a qualitative comparison of the arms involved in the performed action. In the following, we therefore discuss all possible types of feedback, throughout assuming that $A_t \in \mathbb{A}$ is the action of the learner in iteration t, and $I(A_t) \subseteq [K]$ is the corresponding index set of the arms in A_t.

6.2.1 All Pairwise Preferences

One natural way to leverage the concepts of the dueling bandits to the multi-dueling bandits is by assuming that all (noisy) pairwise preferences among the involved arms in A_t are observed, which in turn are still governed by an underlying (unknown) preference relation $\mathbf{Q}$ specifying the pairwise preference probabilities. Formally, if $|A_t| \geq 2$, the feedback[22] consists of a sequence of length $\binom{|A_t|}{2}$, each element of which is either $\{a_i \succ a_j\}$ or $\{a_i \prec a_j\}$ for some distinct $a_i, a_j \in A_t$, where the probabilistic mechanism generating such a pairwise preference is given by (1):

$$\mathbf{P}(a_i \succ a_j) = q_{i,j}, \quad \mathbf{P}(a_i \prec a_j) = 1 - q_{i,j} = q_{j,i}.$$

6.2.2 The Most Preferred Arm

From a high level point of view, by comparing two specific arms in the dueling bandits problem, the learner obtains as its feedback the most preferred arm among these two. Thus, one can easily generalize the concept underlying this type of feedback to cases with more than two arms involved in the comparison by assuming that the information received is still only the most preferred arm, but now among *all arms* involved in A_t. Formally, the feedback is a partial ranking of the form $a_i \succ A_t \backslash \{a_i\}$ for exactly one $a_i \in A_t$, which is observed with probability

$$q_{i|I(A_t)} := \mathbf{P}(a_i \succ A_t \backslash \{a_i\}). \tag{20}$$

Such type of feedback is of great importance especially in the fields of economics or social sciences and studied there under the notion of *discrete choice models* (Train, 2009). Discrete choice models specify the probability that an individual chooses one alternative among a given set of choice alternatives, which is essentially represented by the left-hand side of (20) by regarding choice alternatives and arms as the same. Formally, a discrete choice model assumes a latent family of categorical distributions $(q_{i|I(A)})_{A \in \mathbb{A}, i \in I(A)}$ for every admissible set of choice alternatives (arms) $A \in \mathbb{A}$.

A popular class of such categorical distribution families is specified by a *Random Utility Model* (RUM), where each arm (choice alternative) $a_i \in \mathcal{A}$ is assumed to be equipped with a (latent) utility $\theta_i > 0$ and the discrete choice probability is given by

$$q_{i|I(A_t)} = \mathbf{P}\left(u_i = \max_{a_j \in A_t} u_j\right), \tag{21}$$

22. In case $|A_t| = 1$, i.e., a "full commitment" to one arm (compare Section 2.3.1), the learner observes obviously no preference information.

where $u_i = \theta_i + \zeta_i$ and $\zeta_1, \ldots, \zeta_K$ is an identically distributed sample of some probability distribution $\mathbf{P}^*$. Apparently, the probability specified by (21) depends on the concrete values of the latent utilities as well as on the distribution $\mathbf{P}^*$ of the noise terms $\zeta_1, \ldots, \zeta_K$. Different parametric probability models are used for the random noise in order to assess the probability in (21), whereas it is common to assume that the noise terms are i.i.d.

In general, the right-hand side of (21) has no closed analytical form and even worse, might be computationally costly. Nevertheless, RUMs still provide a convenient way to facilitate theoretical considerations, as there is a natural ordering of the arms due to the underlying latent utility structure on the one side (cf. Section 3.2), and on the other side, RUMs offer a quite intuitive explanation of the feedback generation: The individual (or nature) first assigns each available choice alternative (arm) a noisy utility $(u_i)_{a_i \in A_t}$, then sorts the available choice alternatives according to their noisy utilities, and finally chooses the one with the highest noisy utility.

Here, the use of noise terms to perturb the actual utilities is reasonable, because external effects such as information asymmetry or impreciseness in the choice mechanism might occur, which can lead to a biased perception of the actual utility of each choice alternative. In other words, the individual (or nature) may not act perfectly according to the latent utilities $\theta_1, \ldots, \theta_K$ of the choice alternatives.

Some popular special cases of a RUM include the following:

- The *multinomial logit (MNL) model*, where $\zeta_1, \ldots, \zeta_K$ is an i.i.d. sample of a standard Gumbel distribution. Quite interestingly, the MNL model admits a closed analytical form for (21) given by

$$q_{i|I(A_t)} = \frac{\exp(\theta_i)}{\sum_{a_j \in A_t} \exp(\theta_j)} \ . \tag{22}$$

- The *multinomial probit (MNP) model*, where $\zeta_1, \ldots, \zeta_K$ is an i.i.d. sample of a standard Gaussian distribution. In contrast to the MNL model, there is in general no closed analytical form for (21) for the MNP model.

Another natural way to come up with a reasonable family of categorical distributions is by assuming a probability distribution $\mathbf{P} : \mathbb{S}_K \to [0,1]$ such as in Section 3.3. The probability of the most preferred arm in (20) can be obtained in the same spirit as in (9) by summing over all rankings $\pi \in \mathbb{S}_K$ in which a_i precedes all arms in $A_t \backslash \{a_i\}$:

$$q_{i|I(A_t)} = \sum_{\pi \in \mathbb{S}_K \, : \, \pi(j) > \pi(i) \, , \forall j \in I(A_t) \backslash \{i\}} \mathbf{P}(\pi) \ , \tag{23}$$

where $\pi(j)$ is the rank of $a_j \in A_t$ with respect to π (smaller ranks indicate higher preference). It is worth noting that the Plackett-Luce (PL) model (cf. Section 3.3.2) coincides with the MNL model in this regard, i.e., if the score parameter $\theta \in \mathbb{R}_+^K$ of the PL model is set to $(\exp(\theta_1), \ldots, \exp(\theta_K))^\top$, then the marginals in (23) of the PL model are equivalent to the right-hand side of (22).

6.2.3 The l' Most Preferred Arms

Another practically relevant type of feedback is an ordered list of $l' \in \{1, \ldots, |A_t|\}$ arms involved in the action at time t. Here, it is reasonable to consider the action space $\mathbb{A}_l$ with $l \geq l'$, or to allow that l' can vary with each iteration if the action space is $\mathbb{A}_{\leq l}$ for instance. Formally, a partial ranking $\pi \in \mathbb{S}_{K|I(A_t)}$ of the form

$$a_{\pi^{-1}(1)} \succ a_{\pi^{-1}(2)} \succ \ldots \succ a_{\pi^{-1}(l')} \succ A_t \backslash \{a_{\pi^{-1}(1)}, a_{\pi^{-1}(2)}, \ldots, a_{\pi^{-1}(l')}\}$$

is observed as the feedback, where $\mathbb{S}_{K|I(A_t)}$ is the set of all permutations restricted to the set $I(A_t)$[23] and $\pi^{-1}(i)$ is the index of the arm having the ith best rank with respect to π.

The two probabilistic approaches for modeling the feedback generation in the previous case can here be used in a similar way:

- *Probabilistic ranking model* — Assuming an underlying probabilistic model for $\mathbf{P}$ on $\mathbb{S}_K$, the probability of obtaining $\pi \in \mathbb{S}_{K|I(A_t)}$ is given by

$$\sum_{\tilde{\pi} \in \mathbb{S}_K(\pi, I(A_t))} \mathbf{P}(\tilde{\pi}) \ ,$$

  where

$$\mathbb{S}_K(\pi, I(A_t)) = \Big\{ \tilde{\pi} \in \mathbb{S}_K \, \Big| \, \forall i \in \{2, \ldots, l'\} : \tilde{\pi}(\pi^{-1}(i)) > \tilde{\pi}(\pi^{-1}(i-1)), \\ \forall j \in I(A_t) \backslash \{\pi^{-1}(1), \ldots, \pi^{-1}(l')\} : \tilde{\pi}(j) > \tilde{\pi}(\pi^{-1}(l')) \Big\}$$

  is the set of all rankings respecting the order of the arms in A_t according to π (smaller ranks indicate higher preference).

- *Discrete choice model* — By making the assumption of an underlying discrete choice model $(q_{i|I(A)})_{A \in \mathbb{A}, i \in I(A)}$, the probability of observing the partial ranking $\pi \in \mathbb{S}_{K|I(A_t)}$ is equal to

$$\prod_{i=1}^{l'} q_{\pi^{-1}(i)|I(A_t) \backslash \{\pi^{-1}(1), \ldots, \pi^{-1}(i-1)\}} \ .$$

  In words, the latter probability is equivalent to the process, in which iteratively the most preferred arm among all "remaining" arms is (stochastically) determined, where in each iteration, the set of remaining arms is reduced by removing the most preferred arm of the previous iteration. Note that this is similar to the stagewise generation process underlying the PL model (cf. Section 3.3.2).

Apparently, if $l' = 1$, then the feedback about the l' most preferred arms and the feedback about the most preferred arm coincide. Consequently, for $l' > 1$, the former type of feedback is more informative than the latter. However, in general it is not possible to compare the information content of the feedback about the l' most preferred arms and the

23. The index set $I(A_t)$ corresponding to the arms in A_t has to be used, as $\mathbb{S}_K$ has been defined as the set of all permutations on $[K] \subset \mathbb{N}$.

feedback in the form of all pairwise preferences even if $l' = |A_t|$. Although it is true that a partial ranking occurring in the latter case can be transformed into a sequence of all pairwise preferences via the technique of *rank-breaking* (Soufiani et al., 2013), it is an open question whether this technique introduces a bias for the pairwise preference estimates[24]. The other way around, a (noisy) sequence of all pairwise preferences might not be aggregated into a consensus ranking of all arms involved in A_t.

6.2.4 Partial Preferences

Finally, the most general type of feedback one may think of is certainly that of a sequence of partial preferences of the arms involved in the learner's action at time t, which opposed to the feedback scenario involving the l' most preferred arms can now be present in any form, i.e., a sequence of k-wise preferences with $k \in \{1, \ldots, |A_t|\}$ or combinations of the latter for different values of k. Once again, an underlying probabilistic model for $\mathbf{P}$ on $\mathbb{S}_K$ or a discrete choice model can be leveraged in order to specify the probability of observing such a partial ranking.

6.3 Learning Tasks

The learning tasks considered in multi-dueling bandits are mostly the same as specified in Section 2.2 for dueling bandits, but in contrast to the latter additionally give rise to the task of finding the *best subset of arms*, which can be seen as a generalization of the task of finding the best arm or finding the top-k arms. Similarly as in the dueling bandits setting, it is oftentimes not obvious how to define a reasonable notion of a best arm, let alone an optimal or best subset of arms in the multi-dueling bandits variant. This issue is once again mainly due to the different types of feedback in the multi-dueling bandits variant, which may suggest different notions of optimality. In the following, we will only discuss the notions of best arm, best subset of arms, and reasonable target rankings.

6.3.1 Best Arm and Ranking of Arms

Clearly, if the underlying feedback mechanism allows one to make inference about the underlying preference relation $\mathbf{Q}$, then the most natural way to define the best arm/the target ranking is to consider the Condorcet winner/total order of the arms, or if its existence is doubtful to opt for alternative notions of a best arm/target ranking as specified in Section 4.1. This is most obviously the case in the feedback scenario, in which all pairwise preferences are observed, but can also be the case for other types of feedback as proposed by Saha and Gopalan (2018) (cf. Section 6.5.4). However, it might be the case that the underlying feedback mechanism is impractical to make inference about the preference relation $\mathbf{Q}$. Then, a workaround to establish the notion of a best arm or a reasonable target ranking can be derived in a similar manner as in Section 3.2 under the assumption of an underlying discrete choice model, or as in Section 3.3 under the assumption of a probabilistic model $\mathbf{P}$ on $\mathbb{S}_K$.

Another reasonable possibility to answer the question of optimality of an arm exists if the feedback mechanism allows inference about the most preferred arm in a subset. In such

24. For the PL model it is known that no bias occurs (cf. Section 6.5.8) due to its independence from irrelevant alternatives (IIA) property (Alvo and Yu, 2014).

cases, one can generalize the concepts of a Condorcet winner or (some of) the alternative notions of best arm in the spirit of Agarwal et al. (2020):

- *Generalized Condorcet winner* — There exists an arm $a_{i^*} \in \mathcal{A}$ such that $q_{i^*|I(A)} > q_{j|I(A)}$ for any $A \in \mathbb{A}$ with $a_{i^*}, a_j \in A$. Here, $I(A) \subseteq [K]$ corresponds to the index set of the arms in $A \in \mathbb{A}$. In words, for any subset it is contained in, the arm a_{i^*} has the highest probability to be the most preferred arm compared to all other arms in that subset.

- *Generalized Copeland winner* — Define the generalized (normalized) Copeland scores of an arm $a_i \in \mathcal{A}$ by means of

$$d_i^{\text{gen}} := \frac{1}{|\mathbb{A}(i)|} \sum_{A \in \mathbb{A}(i)} \mathbb{I}\{\forall a_j \in A \backslash \{a_i\} : \, q_{i|I(A)} > q_{j|I(A)}\} \ ,$$

  where $\mathbb{A}(i) = \{A \in \mathbb{A} : a_i \in A\}$ is the subfamily of sets in $\mathbb{A}$, which contain a_i. Each arm with highest generalized Copeland score is called a generalized Copeland winner.

- *Generalized Borda winner* — Define the generalized sum of expectations or generalized Borda scores by

$$q_i^{\text{gen}} = \frac{1}{|\mathbb{A}(i)|} \sum_{A \in \mathbb{A}(i)} q_{i|I(A)} \ .$$

  Each arm with highest generalized Borda score is called a generalized Borda winner.

It is easy to check that if the action space corresponds to the action space of the basic dueling bandits setting (i.e., $\mathbb{A}_{\leq 2}^{+}$), all these generalized notions coincide with the corresponding dueling bandits variant, because $q_{i|I(A)} = q_{i,j}$ if $A = \{a_i, a_j\}$. However, the major drawback of these generalized concepts is the combinatorial challenge involved in checking the necessary conditions, respectively. Moreover, the drawbacks of the underlying basic variants are still preserved in the sense that the generalized Condorcet winner may not exist and the generalized Copeland/Borda winner may not be unique, but if the generalized Condorcet winner exists, then there is only one generalized Copeland winner, namely the generalized Condorcet winner itself[25]. Nevertheless, the generalized Borda winner can still differ from the generalized Condorcet winner in case it exists, and may still not be unique.

Finally, it is straightforward to define the notions of a generalized Copeland/Borda ranking using the generalized Copeland/Borda scores similarly as in Section 4.1. The natural ranking in the spirit of Section 2.2.2 is to consider a_i to be ranked lower than a_j if $q_{i|I(A)} > q_{j|I(A)}$ for any $A \in \mathbb{A}$ with $a_i, a_j \in A$. Once again, such a ranking may not exist in general, while the generalized Copeland/Borda ranking always exists and agrees/might disagree with the former if it exists[25]. Assuming a RUM for the underlying feedback mechanism, one obtains a natural ranking by sorting the latent utilities, while under the assumption of an underlying probabilistic model for $\mathbf{P}$ on $\mathbb{S}_K$, the target ranking can be defined as in Section 3.3 via the mode of the distribution.

25. This holds for all action spaces listed in Section 6.1.

6.3.2 Best Subset of Arms

In the case of $\mathbb{A} = \mathbb{A}_l$, it is of special interest to define a notion of a best subset of arms. In fact, in this scenario it is quite natural to recommend an optimal action $\mathbb{A}_l$, which in turn is exactly a subset in $\mathcal{A}$ of size l. For this purpose, the notions of reasonable rankings in the previous section can be used in a straightforward way to establish the notion of an optimal subset of arms by setting it equal to the top-l arms according to the corresponding ranking.

However, under the assumption of a RUM with latent utilities $\theta = (\theta_1, \ldots, \theta_K)^\top \in \mathbb{R}_+^K$ for the feedback mechanism, it is also sensible to define the expected utility of an action $A \in \mathbb{A}$ by

$$U_\theta(A) = \sum_{a_i \in A} \theta_i \cdot q_{i|I(A)} \, , \tag{24}$$

where the occurring probabilities above depend on θ as well, see (21). In this way, one may also define the best subset(s) of arms by $A^* \in \mathrm{argmax}_{A \in \mathbb{A}} U_\theta(A)$, i.e., the action(s) with the maximum achievable expected utility. It is worth noting that this best subset is not necessarily composed of the arms with highest utility (cf. Section 6.5.6).

6.4 Performance Measures

Quite unsurprisingly, the performance measures of interest are essentially the same as in Section 2.3, i.e., sample complexity for (ϵ, δ)-PAC learning scenarios and cumulative regret for regret minimization tasks. However, the generality of the multi-dueling bandits allows one to define a reasonable regret per time in an even greater variety. By now, three variants have been considered in the literature for defining the regret per time. These variants shall be reviewed in the following, where we again denote by $A_t \in \mathbb{A}$ the action used in time t.

- *Generalized averaged regret* — If the feedback mechanism allows inference about the pairwise preferences, then this regret is the straightforward extension of the average regret used in (2), i.e.,

  $$r_{t,avg}^{\mathrm{gen}} := \frac{\sum_{a_j \in A_t} \Delta_{i^*,j}}{|A_t|} \quad ,$$

  where i^* is the target arm, e.g., the Condorcet winner assuming it exists. Under the assumption of an underlying RUM with latent utilities $\theta = (\theta_1, \ldots, \theta_K)^\top \in \mathbb{R}_+^K$ for the feedback mechanism, another variant of the generalized averaged regret can be defined quite naturally based on the latent utilities:

  $$r_{t,avg(\theta)}^{\mathrm{gen}} := \frac{\sum_{a_j \in A_t} (\theta_{i^*} - \theta_j)}{|A_t|} \quad ,$$

  where i^* is the index of the arm with the highest utility. The notions of weak and strong regret per time can be generalized in a straightforward way. Similarly as in the dueling bandits case, the generalized averaged regret is only zero if the action in time t is the singleton set $\{a_{i^*}\}$ corresponding to a full commitment to this arm.

- *Regret by shortfall of preference probabilities* — If the feedback mechanism allows inference about the probability that an arm is the most preferred among a subset of arms, then one can consider the cumulated shortfall of the probabilities of all arms in A_t to be the most preferred one if the target arm is also present. Formally,

$$r_{t,sh} := \sum_{a_j \in A_t} q_{i^*|A_t \cup \{a_{i^*}\}} - q_{j|A_t \cup \{a_{i^*}\}} \ ,$$

  where i^* is the target arm, e.g., the (generalized) Condorcet winner assuming it exists.

- *Expected-utility regret* — Recalling the expected utility of an action A_t in (24), the expected-utility regret in time t is then defined as

$$r_{t,\theta} = \max_{A \in \mathbb{A}} U_\theta(A) - U_\theta(A_t).$$

  In words, the difference between the maximum achievable expected utility and the expected utility of the used action.

Finally, it is worth noting that although the feedback in the form of the most preferred arm in the multi-dueling bandits is seemingly more informative than the one in the dueling bandits, there are a couple of "negative" results in the sense that this seemingly larger amount of information does not necessarily lead to a theoretical improvement (in order sense) of the underlying performance measure for the same target. For instance, Ren et al. (2019) show that if $\mathbb{A} = \mathbb{A}_l$ and the feedback is the most preferred arm for each action, then the lower bound on the sample complexity of every algorithm to return the actual total order of the arms (assuming its existence) with confidence $1-\delta$ is the same as for the dueling bandits problem (cf. Section 3.1.21). Other results in this spirit will be discussed in the following section. However, if the feedback corresponds to the l' most preferred arms, the theoretical guarantees are indeed improving with the value of l'.

6.5 Algorithmic Approaches

In the following, we provide an overview of existing methods for the generalized PB-MAB problem. For reasons of clarity, we summarize all approaches in Tables 8 and 9, maintaining the conceptional structure used in the previous sections.

6.5.1 Multileaving

In the general scenario of the online learning-to-rank problem, it is assumed that there exists some set of objects such as web pages and a (possibly infinite) pool of possible ranking models (or rankers) producing a ranked list of the objects, which can be displayed to users. The goal is to find the best ranking model by learning from preference feedback produced by the users. The online learning-to-rank problem can be cast quite naturally as a PB-MAB problem by considering the ranking models as arms. The pairwise preference feedback of two arms as in the dueling bandits problem is then usually the result of an *interleaving* method (Radlinski et al., 2008), which allows one to infer a preference between the two ranking models. Roughly speaking, this is achieved by interleaving the suggested lists of objects, presenting them to a user, and monitoring which object is chosen by a user.

Algorithm	Algorithm class	Action space, feedback and further assumptions	Target(s) and goal(s) of learner	Theoretical guarantee(s)
Multileaving (Section 6.5.1)	Online optimization-based	$\mathbb{A}_l$; Partial rankings; No further explicit assumptions	Satisfactory experimental performance	No theoretical guarantees
Multi-Dueling Bandit (Section 6.5.2)	Generalization (UCB)	$\mathbb{A}_{\text{full}}$; All pairwise preferences; Existence of a Condorcet winner	Expected regret minimization with generalized average regret	No theoretical guarantees
SelfSparring (Section 6.5.3)	Reduction-based	$\mathbb{A}_{\leq l}$; Partial rankings; Total order of arms and approximate linearity	Expected regret minimization with generalized average regret	Asymptotically no-regret
Battling-Doubler (Section 6.5.4)	Reduction-based	$\mathbb{A}_{\leq l}$; Most preferred arm; Linear pairwise subset choice model	Expected regret minimization with generalized average regret	Finite arms: $\mathcal{O}\left(\frac{lK\log^2 T}{\min_{j\neq i^*}\Delta_{i^*,j}}\right)$ resp. $\mathcal{O}\left(l\sqrt{K\,T\,\log^3(T)}\right)$ Infinite arms: $\mathcal{O}\left(\frac{ld^2\log^4 T}{\min_{a\neq a_*}\Delta_{a_*,a}}\right)$
Battling-MulitSBM (Section 6.5.4)	Reduction-based	$\mathbb{A}_{\leq l}$; Most preferred arm; Linear pairwise subset choice model	Expected regret minimization with generalized average regret	$\mathcal{O}\left(\frac{lK\log T}{\min_{j\neq i^*}\Delta_{i^*,j}}\right)$
Battling-Duel (Section 6.5.4)	Reduction-based	$\mathbb{A}^+_{\leq 2}$; Most preferred arm; Pairwise subset choice model	Expected regret minimization with generalized average regret	Same as the used black-box dueling bandit algorithm
MaxMin-UCB (Section 6.5.5)	Generalization (UCB)	$\mathbb{A}_{\leq l}$ & $\mathbb{A}_{l'}$; l' most preferred arms; Underlying MNL model	Regret minimization with 1. Gen. avg. regret 2. Gen. average top-l' regret	1. See (26) 2. See (27)
Thresholding-Random-Confidence-Bound (Section 6.5.6)	Generalization (UCB)	$\mathbb{A}_l$; Most preferred arm; Underlying MNL model	Expected regret minimization with expected-utility regret	$\mathcal{O}(\sqrt{KT\log(T)})$
Confidence-Bound-Racing (Section 6.5.6)	Generalization (Racing)	$\mathbb{A}_{\text{full}}$; Most preferred arm; Underlying MNL model	Expected regret minimization with expected-utility regret	$\mathcal{O}(\sum_{i\neq i^*}\frac{\log(T)}{(\theta_{i^*}-\theta_i)^2})$
Winner Beats All (Section 6.5.7)	Challenge	$\mathbb{A}^+_{\leq l}$; Most preferred arm; Existence of a generalized Condorcet winner	Expected regret minimization with regret by shortfall	$\mathcal{O}\left(\frac{K\log(T)}{\Delta^2_{\text{GCW}}}\right)$ MNL Model: $\mathcal{O}\left(\frac{K\log(T)}{\theta_{i^*}-\theta_i}\right)$

Table 8: Approaches for the multi-dueling bandits problem for regret minimization tasks.

Algorithm	**Algorithm class**	**Action space, feedback and further assumptions**	**Target(s) and goal(s) of learner**	**Theoretical guarantee(s)**
Trace-the-best (Section 6.5.8)	Challenge	$\mathbb{A}_l$; l' most preferred arms; MNL model	(ϵ, δ)-PAC setting for finding best arm	$\mathcal{O}\left(\frac{K}{l'\,\epsilon^2}\log\frac{K}{\delta}\right)$
Divide-and-Battle (Section 6.5.8)	Tournament	$\mathbb{A}_l$; l' most preferred arms; MNL model	(ϵ, δ)-PAC setting for finding best arm	$\mathcal{O}\left(\frac{K}{l'\,\epsilon^2}\log\frac{l}{\delta}\right)$ or (28)
Halving-Battle (Section 6.5.8)	Tournament, Generalization (Successive-Elimination)	$\mathbb{A}_{\leq l}$; Most preferred arm; MNL model	(ϵ, δ)-PAC setting for finding best arm	$\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{1}{\delta}\right)$
PAC-Wrapper (Section 6.5.8)	Tournament	$\mathbb{A}_l$; l' most preferred arms; MNL model	(ϵ, δ)-PAC setting for finding best arm	See (29)
Uniform-Allocation (Section 6.5.8)	Tournament, Generalization (Successive-Elimination)	$\mathbb{A}_l$; l' most preferred arms; MNL model	Small error of probability for finding best arm with fixed budget B for total number of actions	$\mathcal{O}\left(\log n \exp\left(\frac{l' B \Delta_{min}(\theta)}{K + l\log(l)}\right)\right)$ $\Delta_{min}(\theta) = \min_{i \neq i^*} \theta_{i^*} - \theta_i$
Sequential-Pairwise-Battle (Section 6.5.9)	Tournament	$\mathbb{A}_l$; l' most preferred arms; IID-RUM	(ϵ, δ)-PAC setting for finding ranking of arms	$\mathcal{O}\left(\frac{K}{l'\,C^2_{\mathrm{RUM}}\,\epsilon^2}\log\frac{l}{\delta}\right)$
Beat-the-Pivot (Section 6.5.10)	Challenge	$\mathbb{A}_l$; l' most preferred arms; MNL model	(ϵ, δ)-PAC setting for finding ranking of arms	$\mathcal{O}\left(\frac{K}{l'\,\epsilon^2}\log\frac{K}{\delta}\right)$
Score-and-Rank (Section 6.5.10)	Challenge	$\mathbb{A}_l$; Most preferred arm; MNL model	(ϵ, δ)-PAC setting for finding ranking of arms	$\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{K}{\delta}\right)$
AlgPairwise & AlgMulti-wise (Section 6.5.11)	Tournament	$\mathbb{A}_{\leq l}$; Most preferred arm; MNL model	Exact sample complexity for finding top-k ranking	See (30)

Table 9: Approaches for the multi-dueling bandits problem for (ϵ, δ)-PAC learning.

As a matter of fact, the first work on dueling bandits by Yue and Joachims (2009) is also strongly motivated by online learning-to-rank problems.

The multi-dueling bandits counterparts of the interleaving methods are the *multileaving* methods (Schuth et al., 2014, 2015), which extend interleaving methods by allowing to incorporate more than two ranking models at a time, and even more general feedback coming from the user. Based on such multileaving methods, Schuth et al. (2016) propose the Multileave Gradient Descent (MGD) algorithm, which extends DBGD (cf. Section 3.2.1) by first using multiple random directions for the multiwise comparison with the current point, representing the current best ranker, and modifying the update of the current point accordingly to the more general user feedback. To this end, two variants of MGD are suggested, which differ in the way the observed preferences are used for the update: MGD winner takes all (MGD-W) and MGD mean winner (MGD-M). In MGD-W, the current point is updated by sampling uniformly at random one of the directions that were preferred over the current point, and move into this direction with a certain step size. This is basically the straightforward extension of DBGD. The MGD-M algorithm updates the current point by moving into the "mean" direction, i.e., the mean of all directions that were preferred over the current point, again with a pre-specified step size. Oosterhuis et al. (2016) consider the MGD-M algorithm as well, but replace the used multileaving method by Schuth et al. (2016), team draft multileaving, by an extension of the probabilistic interleave method (Hofmann et al., 2011).

Another attempt to improve the practical performance of the DBGD algorithm by means of multileaving methods is suggested by Zhao and King (2016). They construct the Dual-Point Dueling Bandit Gradient Descent (DP-DBGD) algorithm and the Multi-Point Deterministic Gradient Descent (MP-DGD) algorithm, which, similar to MGD, both take multiple directions within one time step into account, but are focused on reducing variances in the underlying gradient approximation of DBGD. For this purpose, DP-DBGD uses two opposite stochastic directions from the current point for the exploration, instead of only one as in DBGD, while exploration in MP-DGD is carried out by using all points lying on the directions determined by the (non-random) standard unit basis vectors with a specific distance from the current point. The update for DP-DBGD corresponds to the winner takes all strategy as in MGD-W, while MP-DGD updates the current point successively into the directions based on the standard unit basis vectors (with a certain step size), which were preferred over the current point. Both algorithms are combined with the proposed multileaving method *Contextual Interleaving*, which takes previous results from the conducted explorations into account to compose the final interleaved list of objects.

As all these algorithms are highly motivated from a practical perspective, they are thoroughly analyzed in various experiments on data sets from the field of information retrieval. Needless to say, all approaches perform superior compared to DBGD in terms of common metrics such as Normalized Discounted Cumulative Gain (NDCG). However, theoretical analyses like for DBGD are not provided for these approaches.

Last nut not least, Pereira et al. (2019) propose the Counterfactual Dueling Bandits (CDB) algorithm inspired by the specific task of music recommendation, where the objects of interest correspond to songs, which can be recommended to a user. However, in their setting the user obtains at each time step only one recommended song instead of a multileaved list of songs, which results in a binary feedback, i.e., whether the recommended song is played until the end or whether it is skipped. Moreover, the recommended song is solely determined by the current point (current ranking model) and a set of candidate songs generated by some retrieval function, but the conceptional idea for updating the current point before the recommendation is made reveals some similarities to the multi-dueling approach as in MGD. More specifically, CDB compares the current point with multiple random directions (exploratory ranking models) based on rewards, which are estimated in a counterfactual manner by considering the rank assigned by the corresponding ranking model to the previously recommended song, and updates the current point successively into the directions (with a specific step size) having a higher estimated reward than the reward of the current point, which is based on the observed binary feedback.

6.5.2 Multi-Dueling

Also inspired by online learning-to-rank problems, Brost et al. (2016) consider the multi-dueling bandits problem with feedback in the form of all pairwise preferences (cf. Section 6.2.1) and the possibility to choose all possible (non-empty) subsets of arms as actions for the learner. Under the assumption that the underlying preference relation $\mathbf{Q}$ has a Condorcet winner, the multi-dueling bandits algorithm (MDB) is proposed, which, based on two types of upper confidence estimates for each pairwise preference probability, maintains a set of optimistic Condorcet winners in the spirit of RUCB, respectively. Here, one of the types is more conservative than the other by means of a multiplicative factor. In case the set of Condorcet winners based on the non-conservative upper confidence estimates is a singleton set, the MDB makes a full commitment to the corresponding arm, and otherwise composes the subset for its action by all arms within the set of Condorcet winners based on the conservative upper confidence estimates. The key idea underlying the usage of the conservative upper confidence estimates is to boost exploration, which in turn should lead to a quick exploitation by means of the non-conservative upper confidence estimates.

It is shown that the algorithm performs well in empirical experiments, if the performance measure is the cumulative regret with the generalized averaged regret for the regret per time. However, the authors do not provide any theoretical guarantees for MDB.

6.5.3 SelfSparring

Allowing the learner to choose a subset of size at most l as its action and assuming feedback in the form of some (though not necessarily all) pairwise preferences of the arms involved in the comparison, Sui et al. (2017b) investigate the multi-dueling bandits problem in two settings: A finite arm scenario with an existing total order of arms and another scenario based on a Gaussian process modeling of the preference probabilities allowing an infinite number of arms. Inspired by the idea of the sparring approach of Ailon et al. (2014b) (cf. Section 5.1), the authors propose the SelfSparring framework, in which l value-based MAB algorithms are used to control the choice of each of the l arms to be drawn in each iteration. For both scenarios, i.e., the utility-based and the one based on Gaussian process, an instantiation using l many Thompson Sampling instances of SelfSparring is suggested: IndSelfSparring and KernelSelfSparring.

In IndSelfSparring, the underlying Thompson sampling algorithms use independent Beta prior and posterior distributions for the arm choice mechanism. Under the assumption of approximate linearity[26] they show that the algorithm converges to the optimal arm with asymptotically optimal no-regret rate of $\mathcal{O}(K \ln(T)/\Delta)$, where the regret is the generalized average regret, up to a constant, and Δ is the calibrated pairwise preference between the best two arms.

The key idea underlying the design of KernelSelfSparring is that $u(a) = \mathbf{P}(a \succ a^*)$, where a^* is the best arm, corresponds to a sample of a Gaussian process $GP(\mu(a), k(a, a'))$ with unknown mean function $\mu : \mathcal{S} \to [0, 1]$ and some appropriate unknown covariance function $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, which needs to be positive semidefinite. Due to the covariance function, it is possible to allow dependencies among the arms in contrast to the previous scenario considered for IndSelfSparring. On account of the Gaussian process modeling, the underlying Thompson sampling algorithms of KernelSelfSparring are using Gaussian process priors and posteriors for the arm choice mechanism.

6.5.4 Battling Bandits

Picking up the dueling bandits metaphor, Saha and Gopalan (2018) refer to the scenario of receiving feedback in the form of the most preferred arm in a multi-dueling bandits setting as the *battling bandits*. They introduce the *pairwise-subset choice model*, which is a probabilistic choice model based on the preference relation $\mathbf{Q}$ for the considered feedback mechanism. In particular, they assume that the probability that an arm a_i is the most preferred among a subset of arms $A \subset \mathcal{A}$ with $|A| = l$ and $a_i \in A$ is

$$\mathbf{P}(a_i \,|\, A) = \sum_{j \neq i : a_j \in A} \frac{2\, q_{i,j}}{l(l-1)} \, . \tag{25}$$

They first propose two algorithms, Battling-Doubler and Battling-MultiSBM, which are generalizations of Doubler and MultiSBM for the dueling bandits case (cf. Section 3.2.3). Further, a third algorithm, Battling-Duel, is suggested, using some dueling bandits algorithm as a black-box in the following way. In each round, the black-box dueling bandits algorithm suggests a pair of arms $(a_{i(t)}, a_{j(t)})$ for the duel, which is then used to define the set A_t for the battle by replicating $a_{i(t)}$ resp. $a_{j(t)}$ for $\lfloor l/2 \rfloor$ resp. $\lceil l/2 \rceil$ or $\lceil l/2 \rceil$ resp. $\lfloor l/2 \rfloor$ times, each with equal probability. With this, the received feedback is either the information that $a_{i(t)}$ is preferred over $a_{j(t)}$ or vice versa, so that this feedback can be propagated to the black-box dueling bandits algorithm. As a consequence, the underlying action space of Battling-Duel is in fact the same as in the dueling bandits problem, while Battling-Doubler and Battling-MultiSBM consider the action space $\mathbb{A}_{\leq l}$.

For the theoretical analysis, the authors assume the existence of a Condorcet winner for $\mathbf{Q}$ and consider the generalized average regret. Under these assumptions and any probability model (25), they show that the Battling-Duel algorithm has the same bound (up to a multiplicative constant) on its expected regret as the underlying black-box dueling bandits algorithm for the cumulative regret in (2). More interestingly, they show that the lower bound under these assumptions is of the same order as in the dueling bandits case and does not decrease with the size of compared arms. This confirms that, from a theoretical

26. This is a generalization of the linear utility-based setting of Ailon et al. (2014b), where for any triplet of arms such that $a_i \succ a_j \succ a_k$ and some constant $\gamma > 0$, it holds that $\Delta_{i,j} - \Delta_{j,k} \geq \gamma \Delta_{i,k}$.

point of view, there is no advantage in comparing subsets of more than two arms for such pairwise-subset choice models if feedback is given in the form of the most preferred arm.

For Battling-Doubler and Battling-MultiSBM, respectively, gap-dependent as well as gap-independent upper bounds on the expected regret are proved under the additional assumption that the pairwise probabilities are given by (5) for the linear link function. However, all these bounds exhibit a leading multiplicative term l (see Table 8) and, consequently, are not optimal with respect to their dependency on l.

6.5.5 MaxMin-UCB

Saha and Gopalan (2019b) assume a PL model with utility parameter θ (cf. Section 3.3.2) for the generation of the feedback in the form of the l' most preferred arms if the action space is $\mathbb{A}_{\leq l}$, where l' is some non-negative integer strictly smaller than l. Under this assumption, the method of *rank-breaking* (Soufiani et al., 2014), in which a ranking is decomposed into all possible pairwise preference relations coherent with the ranking, does not introduce any bias in the empirical estimation of the pairwise preference probabilities. This is due to the independence from irrelevant alternatives (IIA) property of the PL model (cf. Alvo and Yu (2014)).

They suggest the MaxMin-UCB algorithm, which is essentially a generalization of the RUCB algorithm (cf. Section 3.1.3) using only subsets of sizes $l'+1$ or 1 (full commitment to one arm) for its actions. More specifically, MaxMin-UCB maintains also a set of optimistic Condorcet winners[27] based on the upper confidence estimates of the pairwise preference probabilities extracted by the rank-breaking method and composes the subset of size $l'+1$ by first choosing one potential optimistic Condorcet winner and then iteratively adding an arm to the subset until the subset has a size of $l'+1$. Here, the added arm in one iteration corresponds to the overall worst competitor regarding all previously added arms based on the upper confidence estimates. In case the set of optimistic Condorcet winners is a singleton set, the algorithm makes a full commitment to this arm, i.e., chooses this singleton set as its action subset. It is shown that MaxMin-UCB has a regret bound both with high probability and in expectation of order

$$\mathcal{O}\left(\frac{\log T}{l'} \max_{i \neq i^*} \frac{1}{\Delta_i^2(\theta)} \sum_{i \neq i^*} \Delta_i(\theta)\right), \tag{26}$$

if the regret per time is measured by the generalized average regret with latent utilities. Here, $\Delta_i(\theta) = \theta_{i^*} - \theta_i$ denotes the difference between the utilities of the arm a_{i^*} with the largest utility and arm a_i. To complement the theoretical analysis, an asymptotic lower bound of order

$$\Omega\left(\frac{K \log(T)}{l' \min_{i \neq i^*} \Delta_i(\theta)}\right)$$

is shown.

Under the same assumptions as above, but with the action space $\mathbb{A}_{l'}$, the authors then investigate the task of minimizing the (cumulative) generalized top-l' regret, where the

27. The Condorcet winner exists due to the PL model.

regret per time for using action A_t is measured by means of

$$r_{t,avg(\theta)}^{l'} := \frac{\sum_{i=1}^{l'} \theta_{(i)} - \sum_{a_j \in A_t} \theta_j}{|A_t|} \; ,$$

with $\theta_{(1)} \geq \theta_{(2)} \geq \ldots \geq \theta_{(K)}$. For this purpose, they suggest the Rec-MaxMin-UCB algorithm, which differs from MaxMin-UCB as it now chooses a subset of size l' for its action, but also makes use of the iterative subset construction based on the upper confidence estimates in order to compose the subset of the allegedly top-l' arms. This algorithm has a generalized top-l' regret bound both with high probability and in expectation of order

$$\mathcal{O}\left(\frac{\log T}{l'} \sum_{i \geq l'+1}^{K} \frac{\Delta_{(l'),(i)}(\theta)}{\min_{j \in [l'-1]} (q_{(l'),(j)} - q_{(i),(j)})^2} \right), \tag{27}$$

where $q_{(i),(j)}$ corresponds to the (i,j)th entry of the permuted preference relation matrix $\mathbf{Q}$ such that the kth row corresponds to the pairwise preference probabilities of the arm with the kth largest utility. The authors also provide an asymptotic lower bound of order

$$\Omega\left(\frac{(K - l') \log(T)}{l' \, (\theta_{(l')} - \theta_{(l'+1)})} \right),$$

and show that Rec-MaxMin-UCB is asymptotically optimal with respect to l', K and T.

Finally, the authors show that regret by shortfall of preference probabilities (cf. Section 6.4) and the theoretically analyzed generalized average regret are equivalent up to constants if the components of the PL model parameters are elements of a compact interval.

6.5.6 Preselection Bandits

Interpreting the action within a multi-dueling bandits problem as a preselection for a user, who in turn chooses an arm from this subset, Bengs and Hüllermeier (2020) introduce the notion of *preselection bandits*, along with the question what a good preselection of arms should look like. To this end, they assume feedback in the form of the most preferred arm, latent utilities $\theta_i > 0$ for each arm a_i, and introduce the expected utility in (24) to assess the value of a preselection. They further restrict the analysis to the PL model (cf. Section 3.3.2) and model the utility parameters of the form $\theta_i = v_i^{\gamma}$ in order to incorporate the degree of preciseness of a user via the parameter $\gamma > 0$.

For the case $\mathbb{A} = \mathbb{A}_l$, it is shown that the optimal preselection (i.e., actions in $\mathbb{A}$) does not necessarily consist of the arms with highest latent utilities. Instead, it is composed of top and worst arms, while the arm with highest utility is always in the optimal preselection. They propose the Thresholded-Random-Confidence-Bound (TRCB) algorithm, which maintains estimates of the relative utility parameters θ_i/θ_j for distinct $i, j \in [K]$ as well as confidence bounds on the latter ratios, and uses the preselection (subset of arms) having the largest expected utility for randomly sampled values within the (trimmed) confidence intervals of the ratios θ_i/θ_j. It is shown that the algorithm has an expected-utility regret bound of $\mathcal{O}(\sqrt{KT \log(T)})$, which is nearly optimal as they derive a lower bound of $\Omega(\sqrt{KT})$ if $l \leq K/4$. Interestingly, the cardinality l of the preselections does not play a role in these bounds, although the action space has complexity $\Theta(K^l)$, suggesting that the computation of the optimal preselection with maximal expected utility involves a seemingly difficult combinatorial task. However, as shown by the authors, one can derive structural properties on the expected utility that considerably facilitate the latter optimization problem.

For the case $\mathbb{A} = \mathbb{A}_{\text{full}}$, where the optimal preselection (i.e., actions in $\mathbb{A}$) is obviously the singleton set consisting of the arm with highest utility, they suggest the Confidence-Bound-Racing (CBR) algorithm. CBR keeps track of the pairwise preference probability estimates, which are unbiased thanks to the appealing properties of the PL model, and composes the preselections in each iteration by a random process as follows. First, the arm with the highest empirical wins is selected as the current leader and an arm is randomly included into the preselection based on the share of its confidence interval on its pairwise preference probability of being preferred over the current leader above 1/2. The larger the share, the higher the chance of being included, while an arm is excluded from consideration once the entire confidence interval is below 1/2. This algorithm enjoys a bound on its expected-utility regret of $\mathcal{O}(\sum_{i \neq i^*} \frac{\log(T)}{(\theta_{i^*} - \theta_i)^2})$, where i^* denotes the index of the arm with the highest utility. Both a worst-case lower bound of $\Omega(\sqrt{T})$ and an asymptotic problem-dependent lower bound of $\Omega(K \log(T))$ are shown for this problem scenario.

6.5.7 Choice Bandits

Agarwal et al. (2020) consider the multi-dueling setting, where the learner is allowed to choose any non-empty subset of arms of size up to $l \leq K$ and is provided with the feedback of the most preferred arm among the chosen subset. The authors assume the existence of a generalized Condorcet winner and suggest the Winner Beats All (WBA) algorithm, which proceeds in rounds and revolves around finding a suitable champion arm, which ideally should be the generalized Condorcet winner, and quite naturally should be involved in any chosen action subset. To this end, WBA maintains a round-dependent set of possible challengers and chooses in each iteration within each round the champion arm as the one being empirically preferred over the maximum number of challenger arms, which have not been used in the current round. In particular, the champion arm can change several times within one round. The action subset in each iteration within one round is then composed of the current champion arm and an arbitrary subset of size up to $l-1$ of the possible challengers, which have not been used in the current round (possibly also adding already used challenger arms if the set is small). Once all possible challengers have been used in one round, WBA computes the set of potential challengers for the next round by

$$\{a_i \in \mathcal{A} \,|\, \exists A \subset \mathcal{A} : I_{a_i}(t, A) \geq \log(t) + f(K, |A|)\},$$

where f is some appropriately chosen non-negative function, and the empirical divergence $I_{a_i}(t, A)$ of an arm a_i at time t for a subset $A \subset \mathcal{A}$ (similarly as RMED in Section 3.1.7) is defined by

$$I_{a_i}(t, A) = \sum_{\{a_j \in A : \, \widehat{q}^{\,t}_{i,j|j} \leq 1/2\}} n^t_{i,j|j} \; \mathrm{KL}(\widehat{q}^{\,t}_{i,j|j}, 1/2).$$

Here, $\widehat{q}^t_{i,j|j}$ and $n^t_{i,j|j}$ differ from $\widehat{q}^t_{i,j}$ and $n^t_{i,j}$ by conditioning on the event that j was the current champion while a preference of the form $a_i \succ A_t \backslash \{a_i\}$ with $a_j \in A_t$ was observed, in order to compute the respective quantities.

The motivation of using the latter pairwise preference estimates is that unlike in the case of an underlying PL model (or MNL model) the pairwise estimates $\widehat{q}^t_{i,j}$ might be biased, but $\widehat{q}^t_{i^*,j|j}$ are concentrating around a suitable term which is larger than $1/2$ if i^* is the generalized Condorcet winner. As a consequence, the current champion chosen in the above way will quickly become the generalized Condorcet winner and the set of possible challengers will reduce quickly as well. Equipped with this concentration result, the authors show an expected regret bound for WBA of order $\mathcal{O}\left(\frac{K\log(T)}{\Delta^2_{\mathrm{GCW}}}\right)+\mathcal{O}\left(\frac{K^2\log(K)}{\Delta^2_{\mathrm{GCW}}}\right)$, where Δ_{GCW} is some complexity parameter for the most-preferred-arm probabilities in (20) under the assumption of an existing generalized Condorcet winner, and the regret is measured by the shortfall of preference probabilities regret. In the case of an MNL model with utility parameter θ, the expected regret bound can be refined to $\mathcal{O}\left(\frac{K\log(T)}{\Delta_i(\theta)}\right) + \mathcal{O}\left(\frac{K^2\log(K)}{\min_{i\neq i^*}\Delta^2_i(\theta)}\right)$, where $\Delta_i(\theta)$ is as in Section 6.5.5. For their learning scenario, the authors also show an asymptotic lower bound (for any consistent learner) of order $\Omega\left(\frac{K\log(T)}{\Delta_{\mathrm{GCW}}}\right)$, which corresponds to $\Omega\left(\frac{K\log(T)}{\Delta_i(\theta)}\right)$ in the case of the MNL model, so that WBA is order optimal for the MNL model.

6.5.8 Nearly Best Arm of Plackett-Luce Model

Again assuming a PL model and allowing the learner to choose actions in $\mathbb{A}_l$, whereupon the most preferred arm is returned as the feedback, Saha and Gopalan (2019c) analyze the (ϵ, δ)-PAC learning setting of finding the best arm. More specifically, if $\theta_i \in \mathbb{R}_+$ is the (latent) utility of arm a_i according to the PL model, then one seeks to find an arm a_i such that $\Delta_{i^*,i} > -\varepsilon$ for some $\varepsilon \in (0, 1/2)$ with probability at least $1-\delta$, where $i^* \in \mathrm{argmax}_{i\in[K]}\,\theta_i$.

The authors propose two algorithms for this learning scenario. The first one, Trace-the-best, is a rather typical challenge algorithm, which generalizes the idea of Interleaved Filtering (cf. Section 3.1.1) by breaking the most preferred arm feedback into pairwise winner feedback, i.e., the overall most preferred arm is preferred in a direct comparison with each arm involved in the action subset. In each challenge round, the action subset is composed of the current champion and a randomly chosen $(l-1)$-sized subset of active arms (possibly adding non-active arms), and this action subset is chosen for a specific number of iterations depending on the approximation quality ϵ and the confidence level δ. At the end of the round, the current champion is replaced if and only if there is an arm being empirically preferred over the former by a margin $\epsilon/2$. All arms except the possibly replaced champion of the subset become non-active, and Trace-the-best proceeds with the above sampling procedure until only one arm remains active.

The second one, Divide-and-Battle, is a tournament-based algorithm that first divides the set of arms into subsets of size l, and each subset is successively used as the action for a specific number of times (depending on ϵ and δ). Then, all arms within one subset are eliminated except for the arm having the largest number of wins, and Divide-and-Battle proceeds in the same manner as for the first step on the remaining active arms until only a single active arm is left.

Trace-the-best enjoys a sample complexity of order $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{K}{\delta}\right)$ to return an ϵ-best arm with probability at least $1-\delta$, while Divide-and-Battle improves upon the logarithmic term revealing a sample complexity of order $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{l}{\delta}\right)$. It is shown that these bounds are optimal up to the logarithmic term, respectively, as the authors prove that in this learning scenario the lower bound on the sample complexity is the same as for the dueling bandits scenario (cf. Section 3.1.11 and Section 3.1.12). Thus, similarly as for the regret minimization scenarios considered before, the possibility of comparing more than two arms at the same time does not result in a better theoretical performance for feedback in the form of the most preferred arm.

However, the authors also consider the more general feedback scenario of obtaining the l' most preferred arms in each time step. For this kind of feedback, they provide a lower bound of order $\Omega\left(\frac{K}{l'\epsilon^2}\log\frac{1}{\delta}\right)$ for the sample complexity, showing that the learning task is facilitated thanks to the more informative feedback similarly as in the regret minimization scenario in Section 6.5.5. To this end, they extend their algorithms Trace-the-best and Divide-and-Battle for this kind of feedback by using the rank-breaking method and adapting the lengths of the rounds appropriately. Also, Divide-and-Battle is modified by means of its arm elimination strategy after a round: If there is an arm that is empirically preferred over each arm (in a direct pairwise comparison) in its corresponding subset by an ϵ-dependent margin, then this arm proceeds to the next round and all others are eliminated, while if there is no such arm, then a randomly chosen arm proceeds.

Finally, if the action space is $\mathbb{A}_{\leq l}$ and the learner again observes feedback in the form of the most preferred arm, the authors propose Halving-Battle, which is a tournament-based algorithm sharing similarities with the beat-the-mean algorithm (cf. Section 3.1.2). The set of arms is divided into groups of size l and each group is compared for a specific number of times. Subsequently, all arms that have won less times than the empirical median of the number of wins of arms inside the group are discarded, while the other ones are maintained. Once again this procedure is repeated until only a single arm is left, which is likely to be an ε-best arm, due to its high chance of having a higher win count than the empirical group median. It is shown that Halving-Battle has a sample complexity of order $\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{1}{\delta}\right)$, and therefore improves upon Trace-the-best and Divide-and-Battle for the same feedback, but also has a more flexible action space.

In a subsequent work, Saha and Gopalan (2020a) propose an enhanced variant of Divide-and-Battle, which differs in the way how the lengths of each round are defined: While Divide-and-Battle ignores the overall utility of a subset and simply uses each subset a fixed number of times for its action (up to adjustments of δ and ϵ), the enhanced version first estimates the overall utility of a subset via a subroutine and then adapts the length of a round accordingly to the estimated overall utility of the action subset of that round. This results in a refined instance-dependent bound on the sample complexity of order

$$\mathcal{O}\left(\frac{K\max_{A\in\mathbb{A}_l}\sum_{a_i\in A}\theta_i}{l}\max\left\{1,\frac{1}{l'\epsilon^2}\right\}\log\frac{l}{\delta}\right). \tag{28}$$

This enhanced variant is then combined with the sampling procedure of the algorithms discussed below in Section 6.5.10 to an epoch-based tournament algorithm called PAC-Wrapper. In each epoch, first the enhanced Divide-and-Battle algorithm is used to find a good anchor arm. Then, the set of all non-eliminated arms except the anchor arm is divided into subsets of size $l-1$, merged with the anchor arm, and thereafter each of these subsets is successively used as the action subset for a particular number of times based on the epoch. At the end of an epoch, i.e., after all subsets have been used the epoch-specific number of times, all arms within one subset, which are empirically not preferred over the anchor arm up to some ϵ-dependent margin, are eliminated, and the next epoch is started until only one arm remains.

By adapting the epoch-wise approximation qualities and confidence levels appropriately, as well as using the estimation of the overall utility of an action subset, it is shown that PAC-Wrapper is an (ϵ, δ)-PAC learner with sample complexity of order

$$\mathcal{O}\left(\frac{\max_{A\in\mathbb{A}_l}\sum_{a_i\in A}\theta_i}{l}\sum_{i\neq i^*}\max\left\{1, \frac{1}{l'\max\{\Delta_i^2(\theta),\epsilon^2\}}\right\}\log\frac{l}{\delta}\left(\log\frac{1}{\max\{\Delta_i(\theta),\epsilon\}}\right)\right), \tag{29}$$

where $\Delta_i(\theta)$ is as in Section 6.5.5. Moreover, it is shown that PAC-Wrapper returns the arm with largest latent utility with probability at least $1-\delta$ with a sample complexity as in (29) for ϵ set to 0. Further, a lower bound of order

$$\Omega\left(\max\left\{\frac{1}{l'}\sum_{i\neq i^*}\frac{\theta_i\theta_{i^*}}{\Delta_i^2(\theta)}, \frac{K}{l}\log(1/\delta)\right\}\right)$$

for this learning scenario is derived, showing that PAC-Wrapper has a nearly optimal sample complexity.

Finally, the authors also consider the learning scenario in which the learner has only a limited budget available for the total number of its actions. Such learning scenarios have been considered in the MAB problem (Audibert et al., 2010), but have not received much attention in the PB-MAB problem yet. For this problem scenario, they propose the Uniform-Allocation algorithm, which is conceptionally similar to Halving-Battle, but adapts the lengths of each round with respect to the available budget, and also incorporates that the action space is $\mathbb{A}_l$ as well as that feedback is provided in the form of the l' most preferred arms. The following bound on its probability of error for returning the best arm is shown:

$$\mathcal{O}\left(\log n \exp\left(\frac{l'B\Delta_{min}(\theta)}{K+l\log(l)}\right)\right),$$

where $\Delta_{min}(\theta)=\min_{i\neq i^*}\Delta_i(\theta)$ and B is the available budget. However, the authors also derive a lower bound of $\Omega(\exp(-2l'BH(\theta)))$, where $H(\theta)=\left(\sum_{i\neq i^*}\frac{\theta_i^2}{\Delta_i^2(\theta)}\right)^{-1}$ is a complexity parameter of the underlying PL problem instance, showing that there is a gap between the error bound of Uniform-Allocation and the derived lower bound.

6.5.9 Nearly Best Arm of RUMs

Considering the same learning scenario as in Saha and Gopalan (2019c), but relaxing the PL model assumption[28] to a more general RUM assumption with identically distributed

28. The discussion below (23) justifies to speak of a relaxation.

noise variables (IID-RUM), Saha and Gopalan (2020b) analyze the Divide-and-Battle algorithm under the name Sequential-Pairwise-Battle. To this end, they introduce the notion of *minimum advantage ratio* (Min-AR) of an arm a_i over an arm a_j, which is defined as

$$\min_{A \in \mathbb{A}_l :\, a_i, a_j \in A} \frac{q_{i|I(A)}}{q_{j|I(A)}}\,.$$

By showing a lower bound on the Min-AR for any IID-RUM, the authors derive a condition under which the use of the rank-breaking method, for updating the pairwise preference estimates after receiving feedback in the form of the l' most preferred arms, can still lead to appropriate estimates in the sense that it is still possible to infer which arm has the larger utility parameter. More specifically, if for any pair of arms a_i, a_j, the Min-AR of a_i over a_j is larger than $1 + \frac{4C_{\mathrm{RUM}}(\theta_i - \theta_j)}{C_{\mathrm{RUM}}}$, where $C_{\mathrm{RUM}} > 0$ is some constant depending on the distribution of the noise variables in the underlying IID-RUM, then Sequential-Pairwise-Battle returns the ϵ-best arm with probability at least $1-\delta$ and has a sample complexity of order $\mathcal{O}\left(\frac{K}{l'\, C_{\mathrm{RUM}}^2\, \epsilon^2} \log \frac{l}{\delta}\right)$ if the feedback comes in the form of the l' most preferred arms. Here, the round-wise approximation qualities of Divide-and-Battle have to be modified by incorporating the constant C_{RUM}, giving rise to the Sequential-Pairwise-Battle algorithm. The authors also provide a lower bound for any (ϵ, δ)-PAC learning algorithm for an IID-RUM (fulfilling a qualitatively similar lower bound on its minimum advantage ratios as above) of order $\mathcal{O}\left(\frac{K}{l'\, C_{\mathrm{RUM}}^2\, \epsilon^2} \log \frac{1}{\delta}\right)$, showing that Sequential-Pairwise-Battle is optimal up to logarithmic terms.

6.5.10 Nearly Best Ranking of Plackett-Luce Model

Saha and Gopalan (2019a) investigate the PAC setting of finding an ϵ-best ranking, that is, a ranking π such that $\theta_{\pi^{-1}(k)} \geq \theta_{\pi^{-1}(j)} - \epsilon$ holds for any $1 \leq j < k \leq K$, under the same scenario as in (Saha and Gopalan, 2019c) (cf. Section 6.5.8). For this setting, the authors suggest the Beat-the-Pivot and the Score-and-Rank algorithm, which are both making use of a subroutine Find-the-Pivot in order to obtain a "good" initial anchor arm for eventually inferring the final ranking. The Find-the-Pivot subroutine is an (ϵ, δ)-PAC algorithm for finding the best arm (the same problem as considered in Section 6.5.8) and is just the Trace-the-best algorithm. Both Beat-the-Pivot as well as Score-and-Rank use the same sampling procedure, but differ in the way how the final ranking is returned. More specifically, Beat-the-Pivot uses the estimated pairwise preference probabilities with respect to the anchor arm, say A, while Score-and-Rank uses the estimated relative utility parameters θ_i/θ_A to infer the ranking. The sampling procedure of both algorithms is simply to divide all arms except the anchor arm into subsets of size $l-1$, merge each subset with the anchor arm and thereafter use each of these subsets successively as the action subset for a particular number of times based on δ and ϵ.

Both algorithms have a sample complexity of order $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{K}{\delta}\right)$, which is shown to be optimal by providing a corresponding lower bound. Once again, there is no improvement of the sample complexity compared to the dueling bandits problem (see Table 2). However, assuming feedback in the form of the l' most preferred arms leads to a lower bound of order $\Omega\left(\frac{K}{l'\,\epsilon^2} \log \frac{K}{\delta}\right)$, which Beat-the-Pivot again achieves by exploiting the appealing properties of the PL model via the rank-breaking method.

6.5.11 Exact Top-k Arms of Plackett-Luce Model

Allowing the learner to choose subsets with a size of up to l as its actions, where $l \leq K$ is fixed, and assuming that the feedback in the form of the most preferred arm among the chosen subset is generated by a PL model (or MNL model), Chen et al. (2018) study the problem of identifying the top-k arms. To this end, they consider two regimes for l, one in which $l = \mathcal{O}(\log K)$ and another one for $l = \Omega(\log K)$.

For the logarithmic regime, the authors suggest the AlgPairwise algorithm, which uses only pairs of arms as its action subset, and argue that this restriction to pairs results in a logarithmic term of K in the resulting theoretical guarantee. AlgPairwise maintains a set of active arms and builds the set of top-k arms successively by proceeding in rounds, each of which eliminates active arms by checking whether an arm belongs to the set of top-k arms or an arm belongs to the worst $K-k$ arms with a certain confidence. In each round, the algorithm samples uniformly at random $\tilde{\mathcal{O}}(n)$ many pairs of active arms for its action and computes confidence intervals for the pairwise preference probabilities to decide about the order of their underlying latent scores. Because the order for a specific pair, say a_i and a_j, can not necessarily be decided solely on the basis of the comparisons made between a_i and a_j, the algorithm exploits some sort of transitivity to incorporate order relations between other active arms: If there exist active arms $a_{i_1}, \ldots, a_{i_d}$ such that, based on the comparison made, a_i has a higher (lower) utility than a_{i_1}, a_j has a lower (higher) utility than a_{i_d}, and a_{i_s} has a higher (lower) utility than $a_{i_{s-1}}$ for $s = 1, \ldots, d-1$, then it is inferred that a_i has a higher (lower) utility than a_j. In light of this, an active arm is eliminated and added to the set of top-k arms in case it has a higher utility than $k - k'$ many active arms, while if an active arm has a lower utility than $m - k'$ many active arms, it is simply removed from the set of active arms. Here, k' is the number of arms that are already added to the set of top-k arms, and m is the number of active arms.

It is shown that the set of top-k arms is returned with high probability, while the number of pairwise comparisons is of order

$$\tilde{\mathcal{O}}\Bigg(\frac{K}{l} + k + \frac{\sum_{i=k+1}^{K} \theta_{(i)}}{\theta_{(k)}} + \sum_{i \geq k+1 \colon \theta_{(i)} > \theta_{(k)}/2} \frac{\theta_{(k)}^2}{(\theta_{(k)} - \theta_{(i)})^2} + \sum_{i \leq k \colon \theta_{(i)} \leq 2\theta_{(k+1)}} \frac{\theta_{(k+1)}^2}{(\theta_{(k+1)} - \theta_{(i)})^2} \Bigg), \tag{30}$$

where polylogarithmic terms of K and $(\theta_{(k)} - \theta_{(k+1)})^{-1}$ are hidden in the latter $\tilde{\mathcal{O}}$ term, respectively. For the superlogaritmic regime, the authors suggest the AlgMulti-wise algorithm, which has a sample complexity (i.e., total number of actions) of the same order as in (30). Moreover, the authors derive a lower bound of the same order (without the polylogarithmic terms), showing that the algorithms are nearly optimal.

6.6 Related Frameworks

As mentioned at the beginning of this section, there are some other frameworks that share similarities with the multi-dueling bandits regarding the qualitative nature of the feedback observed by the learner. In the following, we give a brief overview of such frameworks, explaining similarities but also highlighting differences.

6.6.1 Stochastic Click Bandits

The online learning-to-rank problem as described in Section 6.5.1 covers web search as one of its most prominent fields of application. However, the modeling approach by means of the multi-dueling bandits disregards in some way how the user usually interacts with a displayed list of objects such as web pages or documents: The user scans the list from top to bottom, so that the order in which the objects are presented may play a central role. Moreover, the user might not choose any of the presented objects at all.

In stochastic click bandits, these specialties are taken into account by means of a click model, which is a stochastic model of the choice (or click) behavior over an ordered set. A commendable introduction to this setting (and beyond that) can be found in Chapter 32 of Lattimore and Szepesvári (2020), with a thorough overview of the literature on this topic. Nevertheless, it is important to note that multi-dueling bandits and stochastic click models are quite different, mainly due to their different views on the importance on the order of the arms in the subset, and neither one can be considered a special case of the other.

6.6.2 Dynamic assortment optimization

The dynamic assortment optimization problem (Caro and Gallien, 2007) is motivated by the area of retailing, where a retailer seeks to find an optimal assortment (subset) of her/his available products (arms) by repeatedly offering different assortments to a customer, and observe a/no purchase of one product within the offered assortment. Quite naturally, the goal is to maximize the expected revenue over time, which can be expressed as a regret minimization task. This problem framework differs from the multi-dueling bandits in two important points. First, the arms in the former are assumed to be equipped with a priori known revenues, while in the latter, such revenues are simply not present. As a consequence, the optimal action subset might be different in both settings. Second, the customer in the dynamic assortment optimization problem may refuse the purchase. Technically, in the setting of multi-dueling bandits, this can be modeled by extending the set of arms by means of a dummy arm representing this no-choice option and adding this dummy arm to all possible action subsets. While this seems to be an interesting direction for future work, it would require a rethinking of many of the learning tasks and related performance measures.

6.6.3 Best-of-k bandits

In the setting of best-of-k bandits, the learner is allowed to use a k-sized subset of arms as an action (similarly as in multi-dueling bandits with action space $\mathbb{A}_k$) and obtains specific forms of feedback. In contrast to multi-dueling bandits, feedback is usually assumed to be numerical. However, Simchowitz et al. (2016) consider another type of feedback, called *marked-bandit feedback*, in which the learner observes the index of the arm with the largest (latent) reward if it is among the action subset, and otherwise a “void” information. As the rewards are assumed to be binary, the index is chosen uniformly at random from the index set of all arms with positive rewards. Although this type of feedback is once again of a qualitative nature, the "void" information is conceptionally similar to a no-choice option in the dynamic assortment optimization problem. It is an interesting question whether the best-of-k bandits with marked-bandit feedback can be formulated as a specific learning scenario within the multi-dueling bandits by incorporating a dummy arm.

7. Applications

Multi-armed bandits have been used in various fields of application, and the more recent setting of dueling bandits is receiving increasing attention from a practical perspective, too. In the following, we provide a short overview of some recent applications of dueling bandits algorithms.

Chiu and Marsella (2012) consider the problem of learning a model of gesture generation to automatically generate animations for dialogues. To this end, they make use of subjective human judgment of naturalness of gestures. In this regard, pairwise comparisons (one gesture is considered more natural than another one) appear to be much easier than absolute judgments, which are often very noisy. This is why the authors tackle the task as a dueling bandits problem. Concretely, they use the DBGD algorithm (cf. Section 3.2.1) and show empirically that the framework can effectively improve the generated gestures based on the simulated naturalness criterion.

In the context of information retrieval, Hofmann et al. (2013) investigate to what extent historical data in the form of interaction data can lead to an improvement for learning in online learning-to-rank problems. They introduce an approach based on the DBGD algorithm using the Probabilistic Interleave (PI) method for interleaving (cf. Section 6.5.1). They evaluate the performance of their approach in terms of the discounted cumulative reward on several learning-to-rank data sets and find that historical data can indeed be useful to enhance the performance of a learner in online learning-to-rank problems.

Supporting clinical research that aims at recovering motor function after severe spinal cord injury, Sui and Burdick (2014) set up a dueling bandits instance to help paralyzed patients regain the ability to stand on their own feet. The feedback consists of a stochastically ranked subset of K tests, each of which corresponds physically to an electrical stimulation period applied to the spinal cord with a specific stimulus. The goal is to identify the optimal stimulus for a patient, and the ranking is based on a combined scoring of certain standing criteria by the observing clinicians (under noisy conditions). The authors introduce the Rank-Comparison algorithm, a modified version of BTM (cf. Section 3.1.2).

Sui et al. (2017a) address the same application. To overcome the issue of very large action spaces, which is due to the huge number of different stimulating configurations and hinders a fast convergence of algorithms attempting to solve this problem, they consider correlation structures on the action space and exploit dependencies among the arms. This allows them to update a set of active arms instead of only a pair in each iteration of the algorithm. The authors propose CorrDuel, an algorithm based on BTM. This algorithm is applied in a synthetic experimental setup and shown to perform better than algorithms that do not exploit correlation information. In a live clinical trial of therapeutic spinal cord stimulation, CorrDuel performs as good as specialized physicians.

Sui et al. (2018a) consider the safe Bayesian optimization problem, where the goal is to optimize an unknown utility function with absolute or preference feedback in a sequential manner, subject to some unknown safety constraints, in the case where a small region of the safe action space is known a priori and needs to be expanded in an iterative manner. They propose the StageOpt algorithm, which models both the safety and utility functions via Gaussian processes (GPs), and makes use of confidence bounds around the mean function for the sake of (safe) exploration and optimization. Moreover, StageOpt proceeds in two stages for the underlying optimization problem; in the first stage, the safe region is gradually expanded by means of the confidence bounds, while in the second stage, this expanded safe region is used as the domain for Bayesian optimization. For the case of preference feedback, in which the algorithm receives feedback in the form of a Bernoulli distribution with probability as in (5) for the current and the previous sample points (infinite action space), they use the multi-dueling bandit algorithm KernelSelfSparring (cf. Section 6.5.3). Motivated by the clinical application setting described in Sui and Burdick (2014), they consider the goal of finding effective stimulation therapies for patients with severe spinal cord injuries without introducing undesirable side effects, and use the preference-based version of their algorithm to safely optimize clinical spinal cord stimulation in order to help the patients regain physical mobility. They show that the therapies it suggests outperforms the ones proposed by experienced physicians.

Sokolov et al. (2016) use the Structured Dueling Bandits algorithm, an extension of DBGD, for response-based structured prediction in Statistical Machine Translation (SMT). In a repeatable generate-and-test procedure, the learner is given partial feedback that consists of assessments of the quality of the predicted translation, which is used by the learner to update specific model parameters. In a simulation experiment, the authors show that learning from such type of feedback can indeed facilitate the supervision problem and allows a direct optimization of SMT for different tasks.

Chan et al. (2016) consider the problem of the allocation of assessment tasks among peers when grading open-ended assignments in Massive Open Online Courses (MOOCs), and formalize it as a sequential noisy ranking aggregation problem. More specifically, each student ranks a subset of his peers' assignments, and the goal is to aggregate all the partial rankings into a complete ranking of all assignments. The authors assume the existence of a ground-truth ranking, and that the underlying distribution is a Mallows model. Based on these assumptions, they propose TustAwareRankingBasedMAB, an algorithm based on merge sort and MallowsMPR (cf. Section 3.3.1).

Schneider and Kummert (2017) investigate the problem of learning a user's task preferences in Human-Robot Interaction for socially assistive tasks. Concretely, they consider the goal of learning a user's exercise category. They formulate a dueling bandits problem, where arms represent exercises. Preference feedback is given by a user who, given two exercises that are presented to him as text on a display, selects the more preferred one. DTS (cf. Section 4.2.3) is used as a dueling bandits learning algorithm. The simulation experiments show that the users were satisfied with the suggested preference rankings. Moreover, the results of a comparison of the preference learning approach against a simulated strategy that randomly selects preference rankings show that the preference learning approach leads to a significant reduction of ranking errors.

Inspired by the application of MAB algorithms for implementing tree search policies in Monte Carlo tree search (Browne et al., 2012), such as UCT (Upper Confidence bound for Trees), Joppen et al. (2018) introduce the Preference-Based Monte Carlo Tree Search (PB-

MCTS) framework, where the feedback is of a qualitative nature (e.g., ordinal rewards). In contrast to classical tree policies, where a single successor is chosen at each node, resulting in an observation over a path, PB-MCTS triggers two roll-outs at each node, resulting in an observation along a binary tree. Apparently, this variant induces an exponential growth in the number of explored paths. The authors propose a tree policy with a choice mechanism for the successor guided by the RUCB algorithm (cf. Section 3.1.3) and investigate its performance for the 8-puzzle problem.

For the task of automatically recommending suitable parameter settings for an algorithm or solver for sequentially arriving problem instances, El Mesaoudi-Paul et al. (2020) propose the Contextual Preselection with Plackett-Luce (CPPL) algorithm. Here, each algorithm parametrization corresponds to an arm having a latent utility parameter function, which depends on the features of the algorithm parametrization and varies with the features of a problem instance. It is assumed that for an incoming problem instance multiple parametrizations of a solver can be run in parallel for solving the instance, and the parallel solving process is stopped as soon as one of the parametrizations has found a solution. CPPL maintains a pool of candidate parametrizations generated by a genetic crossover procedure using estimates of the utility parameter function in order to decide which algorithm parametrizations are suitable candidates. The candidates with the largest upper confidence on the utility parameter function evaluated for the current problem instance are used for the parallel solving process, and candidates within the pool are pruned by adopting a racing strategy (Maron and Moore, 1994, 1997), whereupon the pool is replenished with new candidates by the genetic crossover procedure. On a series of data sets corresponding to mixed-integer programming and satisfiability problems, the suggested algorithm reveals satisfactory empirical performance in choosing the configurations of corresponding solvers in a sequential manner and even outperforms state-of-the-art methods.

8. Summary and Perspectives

In this paper, we surveyed the state of the art in preference-based online learning with bandit algorithms, a relatively recent research field that we referred to as preference-based multi-armed bandits (PB-MAB), and which is also known under the notion of "dueling bandits". In contrast to standard MAB problems, where bandit information is understood as (stochastic) real-valued rewards produced by the arms the learner decided to explore (or exploit), feedback is assumed to be of a more indirect and qualitative nature in the PB-MAB setting. This includes preference information in the form of comparisons between pairs of arms, which has been the focus of most approaches so far, but which has been extended toward more general feedback scenarios such as partial rankings in the recent past. We have given an overview of instances of the PB-MAB problem that have been studied in the literature, algorithms for tackling them, and criteria for evaluating such algorithms. Besides, we have given an overview of existing applications.

In spite of a growing body of literature, the field is still developing and certainly much less mature than research on standard multi-armed bandits. Obviously, this is due to its recency but also because the setting itself is more involved (for example, inconsistencies such as preferential cycles may occur, the definition of regret is less obvious than in the standard value-based setting, etc.). Various algorithms have been proposed so far, and different theoretical results have been produced. However, comparing these results and relating them to each other is far from obvious, mainly because different authors start from different assumptions and formalizations of the problem; in fact, there is no complete agreement on assumptions, targets, and performance measures so far, and a complete and coherent theoretical framework is still to be developed. Moreover, even for the standard setting, there is still a number of "gaps" in the PB-MAB landscape, i.e., open theoretical questions and algorithmic problems that have not yet been addressed.

With this survey, we hope to contribute to the further popularization, development, and shaping of the field. We conclude the paper with a short (and certainly non-exhaustive) list of open problems that we consider particularly interesting for future work.

For some of the settings and related learning tasks discussed in the paper, the lower bounds are still not known. For instance, the lower bound on the regret for general tournament solutions in Section 4.2.5 has been left as an open question. Also for the task of weak regret minimization considered by Chen and Frazier (2017) (cf. Section 3.1.8), there seems to be still no proven lower bound. Thus, it is difficult to say whether the suggested algorithms are optimal with regard to the problem-dependent complexity terms or not. In the context of statistical approaches (such as Mallows and Plackett-Luce), the analysis of lower bounds would also be interesting for the problem of approximating the entire distribution $\mathbf{P}$. Busa-Fekete et al. (2014b) address this problem for the Mallows model, using the Kullback-Leibler (KL) divergence as a measure of distance between $\mathbf{P}$ and the learner's prediction $\widehat{\mathbf{P}}$. The problem turned out to be hard, and the sample complexity of the authors' algorithm scales quite poorly with the number of arms. Thus, one may conjecture that a truly efficient algorithm does not exist, which in turn calls for the proof of a lower complexity bound.

Next, as we have seen, the difficulty in the basic PB-MAB learning scenario strongly depends on the assumptions on the preference relation $\mathbf{Q}$: The more restrictive these assumptions are, the easier the learning task becomes. Take the (ϵ, δ)-PAC learning scenario for finding a suitable ranking of the arms for instance, for which the results of Falahatgar et al. (2018) show that strong stochastic transitivity and the stochastic triangle inequality, and consequently the exploitation of both properties is the key to achieve a sample complexity which is sub-quadratic with respect to the number of arms. However, in the multi-dueling bandit problem, such transitivity properties are not readily available. It would be an interesting question whether one could define properties giving a full-fledged substitute without imposing too strict requirements such as latent utilities of the arms or suchlike.

Related to the previous issue, another important problem concerns the development of (statistical) tests for verifying the assumptions made by the different approaches in a real application. In the case of the statistical approaches based on the Mallows and PL distributions, for example, the problem would be to decide, based on data in the form of pairwise comparisons, whether the underlying distribution could indeed be Mallows or PL versus a reasonable alternative hypothesis class. In other words, suitable (online) hypothesis tests are needed that allow for deciding whether or not the data-generating process obeys a certain distribution, such as Mallows or Plackett-Luce. Surprisingly little work has been done on this problem in the statistical literature so far (Alvo and Yu, 2014; Fligner and Verducci, 1993). There is some recent work on testing properties of discrete distributions.

For example, given the possibility to sample from a multinomial distribution, the goal is to decide whether the distribution belongs to a family of distributions that is given in advance to the learner, e.g., a family of monotone distributions (Acharya et al., 2015). Similar questions can be addressed in the preference-based setup. For instance, given the possibility to sample pairwise preferences determined by $\mathbf{Q}$, decide whether $\mathbf{Q}$ belongs to some subset of relations with a certain property, such as relaxed or strong stochastic transitivity, satisfying the stochastic triangle inequality, or exhibiting a Condorcet winner as required by many methods. Existing works seem to be restricted to testing the weak stochastic transitivity assumption in an offline setting (Iverson and Falmagne, 1985; Cavagnaro and Davis-Stober, 2014).

The role of adaptivity[29] is not yet fully clear either. In preference-based online learning, we assume that the learner can sample from the underlying pairwise data-generating process, and hence partly control the data to learn from. To what extent does this additional freedom help the learner, facilitate the problem, and possibly improve performance? This question has not been investigated in detail so far. Agarwal et al. (2017) address it in a specific (utility-based) setup, however, it still remains open in most of the cases we discussed in this paper. In particular, the question seems to be hard to answer in the case of the ranking-distribution-based setup. As an important prerequisite, the sample complexity of the optimal learning should be characterized when the learner has sample access to full rankings. For the case of the Mallows model, the sample complexity of optimal learning has been obtained in a recent work by Busa-Fekete et al. (2019). Nevertheless, to the best of our knowledge, there is no known sample-optimal learning algorithm for other parametric ranking distributions, such as Plackett-Luce or log-linear ranking models (El Mesaoudi-Paul et al., 2018).

The Kemeny consensus or minimum feedback arc set (MFAS) ranking can be considered as the holy grail among the ranking methods, because it corresponds to the ranking with the smallest violation of pairwise relations. It is known that if $\mathbf{Q}$ is given, finding the MFAS ranking is NP-hard. There is a constant approximation algorithm (Ailon et al., 2005), and what is more, there is a PTAS for this problem (Kenyon-Mathieu and Schudy, 2007). One can naturally address the question of learning the MFAS ranking with $\mathcal{O}(K \log K)$ sample complexity in the online preference learning setup. In other words, do we need to reveal the whole matrix $\mathbf{Q}$, or might partial information of $\mathbf{Q}$ be enough?

Another interesting variant of the bandit problems themselves is a combination of the preference-based and real-valued MAB problems, such that the learner is allowed to either pull an arm in order to obtain a real-valued reward, or to choose a pair of arms and obtain preference feedback. Such a scenario has recently been considered by Xu et al. (2020) with the goal of finding all arms with a mean reward above some specific threshold, while keeping both the number of duels and pulls as small as possible. The motivation behind such a fusion of both settings is that in some practical applications it might be more difficult to obtain real-valued rewards of arms than a preference between arms. These considerations open up a number of interesting questions, while the main question would be certainly how much the possibility of dueling arms can support the pulling of arms in order to achieve a certain target or vice versa. Even though Xu et al. (2020) already provide some insights and first

29. In the sense of learning in a "non-batch" setup, such as active or online learning, which provides the learner with an opportunity to influence and partly control the training data.

answers to this question, there are definitely more open questions to be addressed in this regard.

Last but not least, there are of course various practically motivated extensions of the basic PB-MAB setting one may think of, along the lines of those summarized in Section 5 and even beyond. In addition to working on such extensions and generalizations, it would be important to test methods and algorithms in real applications, such as crowd-sourcing platforms. However, up to now it seems that there is no available code repository including all (or at least a substantial share) of the established algorithms of the PB-MAB setting. An attempt to address this issue for the programming language Python is made by the *duelpy* package[30].

Acknowledgments

Eyke Hüllermeier, Adil El Mesaoudi-Paul and Viktor Bengs gratefully acknowledge financial support by the German Research Foundation (DFG). We would also like to thank two anonymous referees for their valuable comments and suggestions, which helped to significantly improve this survey.

References

Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 3591–3599, 2015.

Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 39–75, 2017.

Arpit Agarwal, Nicholas Johnson, and Shivani Agarwal. Choice bandits. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 99–107, 2013.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, pages 684–693, 2005.

Nir Ailon, Kohei Hatano, and Eiji Takimoto. Bandit online optimization over the permutahedron. In *Proceedings of the Conference on Algorithmic Learning Theory (ALT)*, pages 215–229, 2014a.

30. `https://gitlab.com/duelpy/duelpy`

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 856–864, 2014b.

Mayer Alvo and Philip L.H. Yu. *Statistical Methods for Ranking Data*. Springer, 2014.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Proceedings of Annual Conference on Learning Theory (COLT)*, 2010.

Julien Audiffren and Liva Ralaivola. Bandits dueling on partially ordered sets. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2129–2138, 2017.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b.

Akshay Balsubramani, Zohar Karnin, Robert E. Schapire, and Masrour Zoghi. Instance-dependent regret bounds for dueling bandits. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 336–360, 2016.

Gábor Bartók. A near-optimal algorithm for finite partial-monitoring games against adversarial opponents. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 696–710, 2013.

Gábor Bartók, Dávid Pál, and Csaba Szepesvári. Minimax regret of finite partial-monitoring games in stochastic environments. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 133–154, 2011.

Gábor Bartók, Dean P. Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring – classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

Viktor Bengs and Eyke Hüllermeier. Preselection bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 778–787, 2020.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 19–26, 2011.

Ralph Alan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Felix Brandt and Felix Fischer. PageRank as a weak tournament solution. In *Proceedings of the International Conference on Internet and Network Economics (WINE)*, pages 300–305, 2007.

Felix Brandt, Markus Brill, Paul Harrenstein, and Hervé Moulin. *Tournament Solutions*, pages 57–84. Cambridge University Press, 2016.

Jürgen Branke, Stephen E. Chick, and Christian Schmidt. Selecting a selection procedure. *Management Science*, 53(12):1916–1932, 2007.

Mark Braverman, Jieming Mao, and S. Matthew Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the ACM symposium on Theory of Computing*, pages 851–862, 2016.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

Brian Brost, Yevgeny Seldin, Ingemar J. Cox, and Christina Lioma. Multi-dueling bandits and their application to online ranker evaluation. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2161–2166, 2016.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1): 1–122, 2012.

Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, pages 18–39, 2014.

Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. Top-k selection based on adaptive sampling of noisy preferences. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1094–1102, 2013.

Róbert Busa-Fekete, Eyke Hüllermeier, and Balázs Szörényi. Preference-based rank elicitation using statistical models: The case of Mallows. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1071–1079, 2014a.

Róbert Busa-Fekete, Balázs Szörényi, and Eyke Hüllermeier. PAC rank elicitation through adaptive sampling of stochastic pairwise preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1701–1707, 2014b.

Róbert Busa-Fekete, Dimitris Fotakis, Balázs Szörényi, and Manolis Zampetakis. Optimal learning of Mallows block model. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, pages 529–532, 2019.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

Felipe Caro and Jérémie Gallien. Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, 53(2):276–292, 2007.

Daniel R. Cavagnaro and Clintin P. Davis-Stober. Transitive in our preferences, but transitive in different ways: An analysis of choice variability. *Decision*, 1(2):102, 2014.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 273–280, 2008.

Hou Pong Chan, Tong Zhao, and Irwin King. Trust-aware peer assessment using multi-armed bandit algorithms. In *Proceedings of International Conference Companion on World Wide Web (WWW)*, pages 899–903, 2016.

Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS)*, pages 2249–2257, 2011.

Bangrui Chen and Peter I. Frazier. Dueling bandits with dependent arms. *CoRR*, abs/1605.08838, 2016.

Bangrui Chen and Peter I. Frazier. Dueling bandits with weak regret. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 731–739, 2017.

Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 379–387, 2014.

Wei Chen, Yihan Du, Longbo Huang, and Haoyu Zhao. Combinatorial pure exploration for dueling bandit. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1531–1541, 2020.

Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM)*, pages 193–202, 2013.

Xi Chen, Yuanzhi Li, and Jieming Mao. A nearly instance optimal algorithm for top-k ranking under the multinomial logit model. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2504–2522. SIAM, 2018.

Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *Proceedings of the International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 51–69, 2007.

Chung-Cheng Chiu and Stacy Marsella. Subjective optimization. In *Proceedings of International Conference on Intelligent Virtual Agents (IVA)*, pages 204–211, 2012.

Haim Cohen and Koby Crammer. Learning multiple tasks in parallel with a shared annotator. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1170–1178, 2014.

William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10(1):243–270, 1999.

Arhur H. Copeland. A reasonable social welfare function. *Seminar on Applications of Mathematics to Social Sciences. University of Michigan, Ann Arbor*, 1951.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 355–366, 2008.

Dotan Di Castro, Claudio Gentile, and Shie Mannor. Bandits with an edge. *CoRR*, abs/1109.2296, 2011.

Miroslav Dudík, Katja Hofmann, Robert E. Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 563–587, 2015.

Adil El Mesaoudi-Paul, Eyke Hüllermeier, and Róbert Busa-Fekete. Ranking distributions based on noisy sorting. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3469–3477, 2018.

Adil El Mesaoudi-Paul, Dimitri Weiß, Viktor Bengs, Eyke Hüllermeier, and Kevin Tierney. Pool-based realtime algorithm configuration: A preselection bandit approach. In *International Conference on Learning and Intelligent Optimization (LION)*, pages 216–232. Springer, 2020.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 255–270, 2002.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Moein Falahatgar, Yi Hao, Alon Orlitsky, Venkatadheeraj Pichapati, and Vaishakh Ravindrakumar. Maxing and ranking with few assumptions. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 7060–7070, 2017a.

Moein Falahatgar, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Maximum selection and ranking under noisy comparisons. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1088–1096, 2017b.

Moein Falahatgar, Ayush Jain, Alon Orlitsky, Venkatadheeraj Pichapati, and Vaishakh Ravindrakumar. The limits of maxing, ranking, and preference learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1426–1435, 2018.

Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

Steffen Finck, Hans-Georg Beyer, and Alexander Melkozerov. Noisy optimization: A theoretical strategy comparison of ES, EGS, SPSA & IF on the noisy sphere. In *Proceedings of Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 813–820, 2011.

Peter C. Fishburn. *Utility Theory for Decision Making*. New York: John Wiley and Sons, 1970.

Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 385–394, 2005.

Michael A. Fligner and Joseph S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):359–369, 1986.

Michael A. Fligner and Joseph S. Verducci. *Probability Models and Statistical Analyses for Ranking Data*. Springer, 1993.

Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer-Verlag, 2011.

Pratik Gajane and Tanguy Urvoy. Utility-based dueling bandits as a partial monitoring game. *CoRR*, abs/1507.02750, 2015.

Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 218–227, 2015.

Aurélien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 359–376, 2011.

Matthew Groves and Jürgen Branke. Top-κ selection with pairwise comparisons. *European Journal of Operational Research*, 274(2):615–626, 2019.

Shengbo Guo, Scott Sanner, Thore Graepel, and Wray Buntine. Score-based Bayesian skill learning. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 106–121, 2012.

Björn Haddenhorst, Eyke Hüllermeier, and Martin Kolb. Generalized transitivity: A systematic comparison of concepts with an application to preferences in the Babington Smith model. *International Journal of Approximate Reasoning*, 119:373–407, 2020.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Reinhard Heckel, Max Simchowitz, Kannan Ramchandran, and Martin Wainwright. Approximate ranking from pairwise comparisons. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1057–1066, 2018.

Reinhard Heckel, Nihar Shah, Kannan Ramchandran, and Martin Wainwright. Active ranking from pairwise comparisons and when parametric assumptions do not help. *The Annals of Statistics*, 47(6):3099–3126, 2019.

Charles A. R. Hoare. Algorithm 65: find. *Communications of the ACM*, 4(7):321–322, 1961.

Charles A. R. Hoare. Quicksort. *The Computer Journal*, 5(1):10–15, 1962.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Katja Hofmann. *Fast and Reliable Online Learning to Rank for Information Retrieval*. PhD thesis, Dutch Research School for Information and Knowledge Systems, Off Page, Amsterdam, 2013.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 249–258, 2011.

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 ACM International Conference on Web Search and Data Mining (WSDM)*, pages 183–192, 2013.

Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 67–79, 2010.

Geoffrey Iverson and Jean-Claude Falmagne. Statistical issues in measurement. *Mathematical Social Sciences*, 10(2):131–153, 1985.

Kevin G. Jamieson and Robert Nowak. Active ranking using pairwise comparisons. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2240–2248, 2011.

Kevin G. Jamieson, Sumeet Katariya, Atul Deshpande, and Robert Nowak. Sparse dueling bandits. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Tobias Joppen, Christian Wirth, and Johannes Fürnkranz. Preference-based Monte Carlo tree search. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 327–340. Springer, 2018.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Shivaram Kalyanakrishnan. *Learning Methods for Sequential Decision Making with Imperfect Representations*. PhD thesis, University of Texas at Austin, Austin, Texas, USA, December 2011.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 227–234, 2012.

Zohar Karnin. Verification based solution for structured MAB problems. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 145–153, 2016.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvári, Claire Vernade, and Zheng Wen. Stochastic rank-1 bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 392–401, 2017.

Sumeet Katariya, Lalit Jain, Nandana Sengupta, James Evans, and Robert Nowak. Adaptive sampling for coarse ranking. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1839–1848, 2018.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Proceedings of International Conference on Algorithmic Learning Theory (ALT)*, pages 199–213, 2012.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

Maurice G. Kendall. *Rank Correlation Methods*. Hafner Publishing Co., 1955.

Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, pages 95–103, 2007.

Johannes Kirschner, Tor Lattimore, and Andreas Krause. Information directed sampling for linear partial monitoring. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 2328–2369, 2020.

Andras Kocsor, Róbert Busa-Fekete, and Sandor Pongor. Protein classification based on propagation on unrooted binary trees. *Protein and Peptide Letters*, 15(5):428–434, 2008.

Pushmeet Kohli, Mahyar Salek, and Greg Stoddard. A fast bandit algorithm for recommendation to users with heterogenous tastes. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 1135–1141, 2013.

Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 1141–1154, 2015.

Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Copeland dueling bandit problem: Regret lower bound, optimal algorithm, and computationally efficient algorithm. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1235–1244, 2016.

Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *CoRR*, abs/1402.6028, 2014.

Wataru Kumagai. Regret analysis for continuous dueling bandit. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1488–1497, 2017.

Wataru Kumagai. Introduction to bandit convex optimization algorithms. In *International Symposium on Information Theory and Its Applications (ISITA)*, pages 36–39, 2018.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms.* Cambridge University Press, 2020.

Chang Li, Ilya Markov, Maarten De Rijke, and Masrour Zoghi. MergeDTS: A method for effective large-scale online ranker evaluation. *ACM Transactions on Information Systems (TOIS)*, 38(4):1–28, 2020.

Chuang-Chieh Lin and Chi-Jen Lu. Efficient mechanisms for peer grading and dueling bandits. In *Asian Conference on Machine Learning (ACML)*, pages 740–755, 2018.

Robert Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis.* Wiley, 1959.

Colin L. Mallows. Non-null ranking models. *Biometrika*, 44(1):114–130, 1957.

Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

John I. Marden. *Analyzing and Modeling Rank Data.* Chapman & Hall, 1995.

Oded Maron and Andrew W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 59–66, 1994.

Oded Maron and Andrew W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 5(1):193–225, 1997.

Lucas Maystre and Matthias Grossglauser. Just sort it! A simple and effective approach to active preference learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 2344–2353, 2017.

Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 2488–2497, 2017.

Hervi Moulin. *Axioms of Cooperative Decision Making.* Cambridge University Press, 1988.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2483–2491, 2012.

Harrie Oosterhuis, Anne Schuth, and Maarten de Rijke. Probabilistic multileave gradient descent. In *Proceedings of European Conference on Information Retrieval (ECIR)*, pages 661–668, 2016.

Guillermo Owen. *Game Theory.* Academic Press, 1982.

Edward Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, 35(1):174–180, 1964.

Erol Peköz, Sheldon M. Ross, and Zhengyu Zhang. Dueling bandit problems. *Probability in the Engineering and Informational Sciences*, page 1–12, 2020.

Bruno Pereira, Alberto Ueda, Gustavo Penha, Rodrygo Santos, and Nivio Ziviani. Online learning to rank for sequential music recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*, pages 237–245, 2019.

Robert L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(1):193–202, 1975.

Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 43–52, 2008.

Siddartha Y. Ramamohan, Arun Rajkumar, and Shivani Agarwal. Dueling bandits: Beyond Condorcet winners to general tournament solutions. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1253–1261, 2016.

Wenbo Ren, Jia Liu, and Ness Shroff. On sample complexity upper and lower bounds for exact ranking from noisy comparisons. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 10014–10024, 2019.

Wenbo Ren, Jia Liu, and Ness Shroff. The sample complexity of best-k items selection from pairwise comparisons. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8051–8072, 2020.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Aadirupa Saha and Aditya Gopalan. Battle of bandits. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Aadirupa Saha and Aditya Gopalan. Active ranking with subset-wise preferences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3312–3321, 2019a.

Aadirupa Saha and Aditya Gopalan. Combinatorial bandits with relative feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 983–993, 2019b.

Aadirupa Saha and Aditya Gopalan. PAC Battling Bandits in the Plackett-Luce Model. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, pages 700–737, 2019c.

Aadirupa Saha and Aditya Gopalan. From PAC to instance-optimal sample complexity in the Plackett-Luce model. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8367–8376, 2020a.

Aadirupa Saha and Aditya Gopalan. Best-item learning in random utility models with subset choices. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4281–4291, 2020b.

Sebastian Schneider and Franz Kummert. Exploring embodiment and dueling bandit learning for preference adaptation in human-robot interaction. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1325–1331, 2017.

Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. Multileaved comparisons for fast online evaluation. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 71–80, 2014.

Anne Schuth, Robert-Jan Bruintjes, Fritjof Buüttner, Joost van Doorn, Carla Groenland, Harrie Oosterhuis, Cong-Nguyen Tran, Bas Veeling, Jos van der Velde, and Roger Wechsler. Probabilistic multileave for online retrieval evaluation. In *Proceedings of the International SIGIR Conference on Research and Development in Information Retrieval*, pages 955–958, 2015.

Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. Multileave gradient descent for fast online learning to rank. In *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM)*, pages 457–466, 2016.

Nihar Shah, Sivaraman Balakrishnan, Aditya Guntuboyina, and Martin Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 11–20, 2016.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Max Simchowitz, Kevin Jamieson, and Benjamin Recht. Best-of-k bandits. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 1440–1489, 2016.

Artem Sokolov, Stefan Riezler, and Tanguy Urvoy. Bandit structured prediction for learning from partial feedback in statistical machine translation. *CoRR*, abs/1601.04468, 2016.

Hossein Azari Soufiani, William Chen, David C. Parkes, and Lirong Xia. Generalized method-of-moments for rank aggregation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, volume 26, pages 2706–2714, 2013.

Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Computing parametric ranking models via rank-breaking. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 360–368, 2014.

Yanan Sui and Joel Burdick. Clinical online recommendation with subgroup rank feedback. In *Proceedings of ACM Conference on Recommender Systems (RecSys)*, pages 289–292, 2014.

Yanan Sui, Yisong Yue, and Joel W. Burdick. Correlational dueling bandits with application to clinical treatment in large decision spaces. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2793–2799, 2017a.

Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017b.

Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with Gaussian processes. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 4788–4796, 2018a.

Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2018b.

Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for Plackett-Luce: A dueling bandits approach. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 604–612, 2015a.

Balázs Szörényi, Róbert Busa-Fekete, Paul Weng, and Eyke Hüllermeier. Qualitative multi-armed bandits: A quantile-based approach. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1660–1668, 2015b.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Kenneth E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2009.

Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 91–99, 2013.

John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1): 295–320, 1928.

Huasen Wu and Xin Liu. Double Thompson sampling for dueling bandits. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 649–657, 2016.

Liyuan Xu, Junya Honda, and Masashi Sugiyama. Dueling bandits with qualitative feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 5549–5556, 2019.

Yichong Xu, Xi Chen, Aarti Singh, and Artur Dubrawski. Thresholding bandit problem with both duels and pulls. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2591–2600, 2020.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1201–1208, 2009.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 241–248, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Tong Zhao and Irwin King. Constructing reliable gradient exploration for online learning to rank. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1643–1652, 2016.

Julian Zimmert and Yevgeny Seldin. Factored bandits. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2835–2844, 2018.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for stochastic and adversarial bandits. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 467–475, 2019.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

Masrour Zoghi, Shimon Whiteson, Maarten de Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM)*, pages 73–82, 2014a.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 10–18, 2014b.

Masrour Zoghi, Zohar Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 307–315, 2015a.

Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *Proceedings of ACM International Conference on Web Search and Data Mining (WSDM)*, pages 17–26, 2015b.