Reducing Dueling Bandits to Cardinal Bandits

Nir Ailon
CS Dept. Technion
Haifa, Israel
nailon@cs.technion.ac.il

Zohar Karnin
Yahoo! Research
Haifa, Israel
zkarnin@gmail.com

Thorsten Joachims
Cs Dept. Cornell
Ithaca, NY
tj@cs.cornell.edu

May 15, 2014

Abstract

We present algorithms for reducing the Dueling Bandits problem to the conventional (stochastic) Multi-Armed Bandits problem. The Dueling Bandits problem is an online model of learning with ordinal feedback of the form "A is preferred to B" (as opposed to cardinal feedback like "A has value 2.5"), giving it wide applicability in learning from implicit user feedback and revealed and stated preferences. In contrast to existing algorithms for the Dueling Bandits problem, our reductions – named **Doubler**, **MultiSBM** and **Sparring** – provide a generic schema for translating the extensive body of known results about conventional Multi-Armed Bandit algorithms to the Dueling Bandits setting. For **Doubler** and **MultiSBM** we prove regret upper bounds in both finite and infinite settings, and conjecture about the performance of **Sparring** which empirically outperforms the other two as well as previous algorithms in our experiments. In addition, we provide the first almost optimal regret bound in terms of second order terms, such as the differences between the values of the arms.

1 Introduction

When interacting with an online system, users reveal their preferences through the choices they make. Such a choice – often termed implicit feedback – may be the click or tap on a particular link in a web-search ranking, or watching a particular movie among a set of recommendations. Connecting to a classic body of work in econometrics and empirical work in information retrieval Joachims et al. (2007), such implicit feedback is typically viewed as an ordinal preference between alternatives (i.e., "A is better than B"), but it does not provide reliable cardinal valuations (i.e., "A is very good, B is mediocre").

1

To formalize the problem of learning from preferences, we consider the following interactive online learning model, which we call the **Utility-Based Dueling Bandits Problem (UBDB)** similar to Yue et al. (2012); Yue & Joachims (2011). At each iteration t, the learning system presents *two* actions $x_t, y_t \in X$ to the user, where X is the set (either finite or infinite) of possible actions. Each of the two actions has an associated random reward (or utility) for the user, which we denote by u_t and v_t, respectively. The quantity u_t (resp. v_t) is drawn from a distribution that depends on x_t (resp. y_t) only. We assume these utilities are in $[0, 1]$. The learning system is rewarded the average utility $U_t^{\mathrm{av}} = (u_t + v_t)/2$ of the two actions it presents, but *it does not observe this reward*. Instead, it only observes the user's binary choice among the two alternative actions x_t, y_t, which depends on the respective utilities u_t and v_t. In particular, we model the observed choice as a $\{0, 1\}$-valued random variable b_t distributed as

$$
\begin{aligned}
\Pr[b_t = 0|(u_t, v_t)] &= \phi(u_t, v_t) \\
Pr[b_t = 1|(u_t, v_t)] &= \phi(v_t, u_t) \,,
\end{aligned}
\tag{1.1}
$$

where $\phi : [0, 1] \times [0, 1] \mapsto [0, 1]$ is a *link* function. Clearly, the link function has to satisfy $\phi(A, B) + \phi(B, A) = 1$. Below we concentrate on linear link functions (defined in Sec. 2). The binary choice is interpreted as a stochastic preference response between the *left alternative* x_t (if $b_t = 0$) and the *right alternative* y_t (if $b_t = 1$). The utility U^{av} captures the overall latent user experience from the pair of alternatives. A concrete example of this UBDB game is learning for web search, where X is a set of ranking functions among which the search engine selects two for each incoming query; the search engine then presents an interleaving Chapelle et al. (2012) of the two rankings, from which it can sense a stochastic preference between the two ranking functions based on the user's clicking behavior.

The purpose of this paper is to show how UBDB can be reduced to the conventional (cardinal) stochastic Multi-Armed Bandit (MAB) problem[1], which has been studied since 1952 Robbins (1952). In MAB, the system chooses only a single action $x_t \in X$ in each round and directly observes its cardinal reward u_t, which is assumed to be drawn from a latent but fixed distribution attached to x_t. The set X in the traditional MAB game is of finite cardinality K. In more general settings Dani et al. (2008); Mannor & Shamir (2011), this set can be infinite but structured in some way. Dani et al. (2008), for example, assume a stochastic setting in which X is a convex, bounded subset of $\mathbb{R}^n$, and the expectation $\mu(x)$ of the corresponding value distribution is $\langle \mu, x \rangle$, where $\mu \in \mathbb{R}^n$ is an unknown coefficient vector and $\langle \cdot, \cdot \rangle$ is the inner product with respect to the standard basis. We refer to this as the *linear expected utility setting*. We study here both the finite setting and the infinite setting.

Main results. We provide general reductions from UBDB to MAB. More precisely, we use a MAB strategy as a black-box for helping us play the UBDB game. The art is in exactly how to use a black-box designed for MAB in order to play UBDB. We present one method, **Doubler** (Section 3) which adds an extra $O(\log T)$ factor to the expected regret function compared to that of the MAB black-box, assuming the MAB black-box

---

[1]One armed bandit is a popular slang for slot machines in casinos, and the MAB game describes the problem faced by a gambler who can choose one machine to play at each instance.

has polylogarithmic (in T) regret, where T is the time horizon. When the MAB black-box has polynomial regret, only an extra $O(1)$ factor is incurred. This algorithm works for infinite bandit spaces. We also present a reduction algorithm **MultiSBM** (Section 4) which works for finite bandit spaces and gives an $O(\log T)$ regret, assuming the MAB black-box enjoys an $O(\log T)$ expected regret function with some mild higher moment assumptions. These assumptions are satisfied, for example, by the seminal UCB algorithm Auer et al. (2002). Our analysis in fact shows that for sufficiently large T, the regret of **MultiSBM** is asymptotically identical to that of UCB not only in terms of the time horizon T but in terms of second order terms such as the differences between the values of the arms; it follows that **MultiSBM** is asymptotically optimal in the second order terms as well as in T. Finally, we propose a third algorithm **Sparring** (Section 5) which we conjecture to enjoy regret bounds comparable to those of the MAB algorithms hiding in the black boxes it uses. We base the conjecture on arguments about a related, but different problem. In experiments (Section 7) comparing our reductions with special-purpose UBDB algorithms, **Sparring** performs clearly the best, further supporting our conjecture.

All results in this extended abstract assume the linear link function (see Section 2), but we also show preliminary results for other interesting link functions in Appendix D.

Contributions in relation to previous work. While specific algorithms for specific cases of the Dueling Bandits problem already exist Yue et al. (2012); Yue & Joachims (2011, 2009), our reductions provide a general approach to solving the UBDB. In particular, this paper provides general reductions that make it possible to transfer the large body of MAB work on exploiting structure in X to the dueling case in a constructive and algorithmic way. Second, despite the generality of the reductions their regret is asymptotically comparable to the tournament elimination strategies in Yue et al. (2012); Yue & Joachims (2011) for the finite case as $T \to \infty$, and better than the regret of the online convex optimzation algorithm of Yue & Joachims (2009) for the infinite case (albeit in a more restricted setting).

In our setting, the reward and feedback of the agent playing the online game are, in some sense, *orthogonal* to each other, or *decoupled*. A different type of decoupling was also considered in Avner et al.'s work Avner et al. (2012), although this work cannot be compared to theirs. There is yet more work on bandit games where the algorithm plays two bandits (or more) in each iteration, e.g. Agarwal et al. Agarwal et al. (2010), although there the feedback is cardinal and not relative in each step. There is much work on learning from example pairs Herbrich et al. (2000); Freund et al. (2003); Ailon et al. (2012) as well as noisy sorting Karp & Kleinberg (2007); Feige et al. (1994), which are not the setting studied here. Finally, our results connect multi-armed bandits and online optimization to the classic econometric theory of discrete choice, with its use of preferential or choice information to recover values of goods (see Train (2009) and references therein).

Another important topic related to our work is that of *partial monitoring games*. The idea was introduced by Piccolboni & Schindelhauer (2001). The objective in partial monitoring is to choose at each round an action from some finite set of actions, and receive a reward based on some unknown function chosen by an oblivious process. The observed information is defined as some (known) function of the chosen action and the current choice of the oblivious process. One extreme setting in which

3

the observed information equals the reward captures MAB. In the other extreme, the observed information equals the entire vector of rewards (for all actions), giving rise to the so-called *full information* game. Our setting is a strict case of partial monitoring as it falls in neither extremes. Most papers dealing with partial monitoring either discuss non-stochastic settings or present problem-independent results. In both cases the regret is lower bounded by $\sqrt{T}$, which is inapplicable to our setting (see Antos et al. (2012) for a characterization of partial monitoring problems). Bartók et al. Bartók et al. (2012) do present problem dependent bounds. Using their work, a logarithmic (in T) bound can be deduced for the dueling bandit problem, at least in the finite case. However, the dependence on the number of arms is quadratic, whereas we present a linear one in what follows. Our algorithms are also much simpler and directly take advantage of the structure of the problem at hand.

2 Definitions

The set of actions (or *arms*) is denoted by X. In a standard stochastic MAB (multi-armed bandit) game, each bandit $x \in X$ has an unknown associated expected utlity $\mu(x) \in [0, 1]$. At each step t the algorithm chooses some $x_t \in X$ and receives from "nature" a random utility $u_t \in [0, 1]$, drawn from a distribution of expectation $\mu(x_t)$. This utility is viewed by the algorithm.[2] The regret at time T of an algorithm is defined as $R(T) = \sum_{t=1}^{T}(\mu(x^*) - u_t)$. where x^* is such that $\mu(x^*) = \max_{x \in X} \mu(x)$ (we assume the maximum is achievable). Throughout, for $x \in X$ we will let Δ_x denote $\mu(x^*) - \mu(x)$ whenever we deal with MAB. (We will shortly make reference to some key results on MAB in Section 2.1.)

In this work we will use MAB algorithms as black boxes. To that end, we define a Singleton Bandit Machine (SBM) as a closed computational unit with an internal timer and memory. A SBM S supports three operations: reset, advance and feedback. The reset operation simply clears its state.[3] The advance operation returns the next bandit to play, and feedback is used for simulating a feedback (the utility). It is assumed that advance and feedback operations are invoked in an alternating fashion. For example, if we want to use a SBM to help us play a traditional MAB game we first invoke reset(S), then invoke and set $x_1 \leftarrow$ advance(S), we will play x_1 against nature and observe u_1 and then invoke feedback(S, u_1). We then invoke and set $x_2 \leftarrow$ advance(S), then we'll play x_2 against nature and observe u_2, then invoke feedback(S, u_2) and so on. For all SBM's S that will be used in the algorithms in this work, we will only invoke the operation feedback($S, \cdot$) with values in $[0, 1]$.

In the utility based dueling bandit game (UBDB), the algorithm chooses $(x_t, y_t) \in X \times X$ at each step, and a corresponding pair of random utilities $(u_t, v_t) \in [0, 1]$ are given rise to, but not observed by the algorithm. We assume u_t is drawn from a distribution of expectation $\mu(x_t)$ and v_t independently from a distribution of expectation $\mu(y_t)$. The algorithm observes a *choice* variable $b_t \in \{0, 1\}$ distributed according to the law (1.1). This random variable should be thought of as the outcome of a duel,

---

[2]It is typically assumed that this distribution depends on x_t only, but this assumption can be relaxed.

[3]We assume the bandit space X is universally known to all SBM's.

or match between x_t and y_t. The outcome $b_t = 1$ (resp. $b_t = 0$) should be interpreted as "y_t is chosen' (resp. "x_t is chosen").[4] The link function ϕ, which is assumed to be known, quantitatively determines how to translate the utilities u_t, v_t to winning probabilities. The linear link function ϕ_{lin} is defined by

$$\Pr[b_t = 1|(u_t, v_t)] = \phi_{\mathrm{lin}}(u_t, v_t) := \frac{1 + v_t - u_t}{2} \in [0, 1] \ .$$

The *unobserved* reward is $U_t^{\mathrm{av}} = (u_t + v_t)/2$, and the corresponding regret after T steps is $R^{\mathrm{av}}(T) := \sum_{t=1}^{T}(\mu(x^*) - U_t^{\mathrm{av}})$, where $x^* = \mathrm{argmax}_{x \in X} \mu(x)$. This implies that expected *zero regret* is achievable by setting $(x_t, y_t) = (x^*, x^*)$. In practice, these two identical alternatives would be displayed as one, as would naturally happen in interleaved retrieval evaluation Chapelle et al. (2012). It should be also clear that playing (x^*, x^*) is pure exploitation, because the feedback is then an unbiased coin with zero exploratory information.

We also consider another form of (unobserved) utility, which is given as $U_t^{\mathrm{choice}} := u_t(1 - b_t) + v_t b_t$. We call this choice-based utility, since the utility that is obtained depends on the user's choice. Accordingly, we define $R_t^{\mathrm{choice}} := \mu(x^*) - U_t^{\mathrm{choice}}$. In words, the player receives reward associated with either the left bandit or the right bandit, whichever was *actually chosen*. The utility U^{choice} captures the user's experience after choosing a result. In an e-commerce system, U^{choice} may capture *conversion*, namely, the monetary value of the choice. Although both utility modelings U^{av} and U^{choice} are well motivated by applications, we avoid dealing with choice based utilities and regrets for the following reason: regret bounds with respect to U^{av} imply similar regret bounds with respect to U^{choice}.

Observation 2.1. *Assuming a link function where $u > v$ implies $\phi(u, v) > 1/2$, for any x_t, y_t, $\mathbb{E}[R_t^{\mathrm{choice}}|(x_t, y_t)] \leq \mathbb{E}[R_t^{\mathrm{av}}|(x_t, y_t)]$.*

(Due to lack of space, the proof can be found in Appendix E.) The observation's assumption on the link function in words is: when presented with two items, the item with the larger utility is more likely to be chosen. This clearly happens for any reasonable link function. We henceforth assume utility U^{av} and regret R^{av} and will no longer make references to choice-based versions thereof.

2.1 Classic Stochastic MAB: A Short Review

We review some relevant classic MAB literature. We begin with the well known UCB policy (Algorithm 1) for MAB in the finite case. The commonly known analysis of UCB provides expected regret bounds. For the finite X case, we need a less known, robust guarantee bounding the probability of playing a sub-optimal arm too often. Lemma 2.2 is implicitly proved in Auer et al. (2002). For completeness, we provide an explicit proof in Appendix A.

---

[4] We have just defined a two-level model in which the distribution of the random variable b_t is determined by the outcome two other random variables u_t, v_t. For simplicity, the reader is encouraged to assume that (u_t, v_t) is deterministically $(\mu(x_t), \mu(y_t))$. Most technical difficulties in what follows are already captured by this simpler case.

Algorithm 1 UCB algorithm for MAB with $|X| = K$ arms. Parameter α affects tail of regret per action in X.

$\forall x \in X$, set $\hat{\mu}_x = \infty$
$\forall x \in X$, set $t_x = 0$
set $t = 1$
while true **do**
    let x be the index maximizing $\hat{\mu}_x + \sqrt{\frac{(\alpha+2)\ln(t)}{2t_x}}$
    play x and update $\hat{\mu}_x$ as the average of rewards so far on action x; increment t_x
    by 1.
    $t \leftarrow t + 1$
end while

Lemma 2.2. *Assume X is finite. Fix a parameter $\alpha > 0$. Let $H := \sum_{x \in X \setminus \{x^*\}} 1/\Delta_x$. When running the UCB policy (Algorithm 1) with parameter α for T rounds the expected regret is bounded by*

$$2(\alpha+2)H\ln(T) + K\frac{\alpha+2}{\alpha} = O(\alpha H \ln T) .$$

Furthermore, lex $x \in X$ denote some suboptimal arm and let $s \geq 4\alpha \ln(T)/\Delta_x^2$. Denote by $\rho_x(T)$ the random variable counting the number of times arm x was chosen up to time T. Then $\Pr[\rho_x(T) \geq s] \leq \frac{2}{\alpha} \cdot (s/2)^{-\alpha}$.

For the infinite case, we will review a well known setting and result which will later be used to highlight the usefulness of Algorithm 2 (and the ensuing Theorem 3.1). In this setting, the set X of arms is an arbitrary (infinite) convex set in $\mathbb{R}^d$. Here, the player chooses at each time point a vector $x \in X$ and observes a stochastic reward with an expected value of $\langle \mu, x \rangle$, for some unknown vector $\mu \in \mathbb{R}^d$.[5] This setting was dealt with by Dani et al. (2008). They provide an algorithm for this setting that could be thought of as linear optimization under noisy feedback. Their algorithm provides (roughly) $\sqrt{T}$ regret for general convex bodies and $\mathrm{polylog}(T)$ regret for polytopes. Formally, for general convex bodies, they prove the following.

Lemma 2.3 (Dani et al. 2008). *Algorithm* CONFIDENCEBALL$_1$ *(resp.* CONFIDENCEBALL$_2$*) of Dani et al. (2008), provides an expected regret of $O\left(\sqrt{dT\log^3 T}\right)$ (resp. $O\left(\sqrt{d^2 T\log^3 T}\right)$) for any convex set of arms.*

In case X is a polytope with vertex set V and there is a unique vertex $v^* \in V$ achieving $\max_{x \in X} \langle \mu, x \rangle$, and any other vertex $v \in V$ satisfies the gap condition $\langle \mu, v \rangle \leq \langle \mu, v^* \rangle - \Delta$ for some $\Delta > 0$, we say we are in the Δ-gap case.

Lemma 2.4 (Dani et al. 2008). *Assume the Δ-gap case. Algorithm* CONFIDENCEBALL$_1$ *(resp.* CONFIDENCEBALL$_2$*) of Dani et al. (2008), provides an expected regret of $O\left(\Delta^{-1} d^2 \log^3 T\right)$ (resp. $O\left(\Delta^{-1} d^3 \log^3 T\right)$).*

---

[5]Affine linear functions can also be dealt with by adding a coordinate fixed as 1.

Algorithm 2 (**Doubler**): Reduction for finite and infinite X with internal structure.

 1: $S \leftarrow$ new SBM over X
 2: $\mathcal{L} \leftarrow$ an arbitrary singleton in X
 3: $i \leftarrow 1, t \leftarrow 1$
 4: **while** true **do**
 5:    reset(S)
 6:   **for** $j = 1...2^i$ **do**
 7:      choose x_t uniformly from $\mathcal{L}$
 8:      $y_t \leftarrow$ advance(S)
 9:      play (x_t, y_t), observe choice b_t
10:     feedback(S, b_t)
11:      $t \leftarrow t + 1$
12:   **end for**
13:   $\mathcal{L} \leftarrow$ the multi-set of arms played as y_t in the last for-loop
14:   $i \leftarrow i + 1$
15: **end while**
---

3   UBDB Strategy for Large or Structured X

In this section we consider UBDB in the case of a large or possibly infinite set of arms X, and the linear link function. The setting where X is large typically occurs when some underlying structure for X exists through which it is possible to gain information regarding one arm via queries to another. Our approach, called **Doubler**, is best explained by thinking of the UBDB strategy as a competition between two players, one controlling the choice of the left arm and the other, the choice of the right one. The objective of each player is to win as many rounds possible, hence intuitively, both players should play the arms with the largest approximated value. Since we are working with a stochastic environment it is not clear how to analyze a game in which both players are adaptive, and whether such a game would indeed lead to a low regret dueling match (see also Section 5 for a related discussion). For that reason, we make sure that at all times one player has a fixed stochastic strategy, which is updated infrequently.

We divide the time axis into exponentially growing epochs. In each epoch, the left player plays according to some fixed (stochastic) strategy which we define shortly, while the right one plays adaptively according to a strategy provided by a SBM. At the beginning of a new epoch, the distribution governing the left arm changes in a way that mimics the actions of the right arm in the *previous* epoch. The formal definition of **Doubler** is given in Algorithm 2.

The following theorem bounds the expected regret of Algorithm 2 as a function of the total number T of steps and the expected regret of the SBM that is used.

Theorem 3.1. *Consider a UBDB game over a set X. Assume the SBM S in Line 1 of* **Doubler** *(Algorithm 2) has an expected regret of $c \log^\alpha T$ after T steps, for all T. Then the expected regret of* **Doubler** *is at most $2c\frac{\alpha}{\alpha+1} \log^{\alpha+1} T$. If the expected regret of the SBM is bounded by some function $f(T) = \Omega(T^\alpha)$ (with $\alpha > 0$), then the expected regret of* **Doubler** *is at most $O(f(T))$.*

The proof is deferred to Appendix B. By setting the SBM S used in Line 1 as the algorithms CONFIDENCEBALL$_1$ or CONFIDENCEBALL$_2$ of Dani et al. (2008), we obtain the following:

Corollary 3.2. *Consider a UBDB game over a set X. Assume that the SBM S in Line 1 of* **Doubler** *is algorithm* CONFIDENCEBALL$_2$ *(resp.* CONFIDENCEBALL$_1$*). If X is a compact convex set, then the expected regret of* **Doubler** *is at most $O(\sqrt{dT\log^3(T)})$ (resp. $O(\sqrt{d^2T\log^3(T)})$). In the Δ-gap setting (see discussion leading to Lemma 2.4), the expected regret is bounded by $O\left(\Delta^{-1}d^2\log^4(T)\right)$ (resp. $O\left(\Delta^{-1}d^3\log^4(T)\right)$).*

In the finite case, one may set the SBM S to the standard UCB, and obtain:

Corollary 3.3. *Consider a UBDB game over a finite set X of cardinality K. Let Δ_i be the difference between the reward of the best arm and the i'th best arm. Assume the SBM S in Line 1 of* **Doubler** *is UCB. Then the expected regret of* **Doubler** *is at most $O(H\log^2(T))$ where $H := \sum_{i=2}^{K}\Delta_i^{-1}$*

Memory requirement issues: A possible drawback of **Doubler** is its need to store the history of y_t from the last epoch in memory, translating to a possible memory requirement of $\Omega(T)$. This situation can be avoided in many natural cases. The first is the case where X is embedded in a real linear space and the expectation $\mu(x)$ is a linear function. Here, there is no need to store the entire history of choices of the left arm but rather the average arm (recall that here the arms are thought of as vectors in $\mathbb{R}^d$, hence the average is well defined). Playing the average arm (as x_t) instead of picking an arm uniformly from the list of chosen arm gives the same result with memory requirements equivalent to storage of one arm. In other cases (e.g., X is not even geometrically embedded) this cannot be done. Nevertheless, as long as we are in a Δ-gap case, as T grows, the arm played as y_t is the optimal one with increasingly higher probability. In more detail, if the regret incurred in a time epoch is R, then the number of times a suboptimal arm is played is at most R/Δ. As R is polylogarithmic in T, the required space is polylogarithmic in T as well. We do not elaborate further on memory requirements and leave this as future research.

4 UBDB Strategy for Unstructured X

In this section we present and analyze an alternative reduction strategy, called **MultiSBM**, particularly suited for the finite X case where the elements of X typically have no structure. **MultiSBM** will not incur an additional logarithmic factor as our previous approach did. Unlike the algorithms in Yue & Joachims (2011); Yue et al. (2012), we will avoid running an elimination tournament, but just resort to a standard MAB strategy by reduction. Denote $K = |X|$. The idea is to use K different SBMs in parallel, where each instance is indexed by an element in X. In step t we choose a left arm $x_t \in X$ in a way that will be explained shortly. The right arm, y_t is chosen according to the suggestion on the SBM indexed by x_t, and the binary choice is fed back to that

Algorithm 3 (**MultiSBM**): Reduction for unstructured finite X by using K SBMs in parallel.

 1: For all $x \in X$: $S_x \leftarrow$ new SBM over X, $\text{reset}(S_x)$
 2: $y_0 \leftarrow$ arbitrary element of X
 3: $t \leftarrow 1$
 4: **while** true **do**
 5:     $x_t \leftarrow y_{t-1}$
 6:     $y_t \leftarrow \text{advance}(S_{x_t})$
 7:     play (x_t, y_t), observe choice b_t
 8:     $\text{feedback}(S_{x_t}, b_t)$
 9:     $t \leftarrow t + 1$
10: **end while**

SBM. In the next round, x_{t+1} is set to be y_t, namely, the right arm becomes the left one in the next step. Algorithm 3 describes **MultiSBM** exactly.

Naively, the regret of the algorithm can be shown to be at most K times that of a single SBM. However, it turns out that the regret is in fact asymptotically competitive with that of a single SBM, without the extra K factor. Specifically, we show that the total regret is in fact dominated solely by the regret of the SBM corresponding to the arm with maximal utility. To achieve this, we assume that the SBM's implement a strategy with a certain robustness property that implies a bound not only on the expected regret, but also on the tail of the regret distribution. More precisely, an inverse polynomial tail distribution is necessary. Interestingly, the assumption is satisfied by the UCB algorithm Auer et al. (2002) (as detailed in Lemma 2.2). Recall that $x^* \in X$ denotes an arm with largest valuation $\mu(x)$, and that $\Delta_x := \mu(x^*) - \mu(x)$ for all $x \in X$. Assume $\Delta_x > 0$ for all $x \neq x^*$.[6]

Definition 4.1. Let T_x be the number of times a (sub-optimal) arm $x \in X$ is played when running the policy T rounds. A MAB policy is said to be α-*robust* when it has the following property: for all $s \geq 4\alpha\Delta_x^{-2}\ln(T)$, it holds that $\Pr[T_x > s] < \frac{2}{\alpha}(s/2)^{-\alpha}$.

Recall that as discussed in Section 2.1, in Auer et al.'s (2002) classic UCB policy this property can be achieved by slightly enlarging the confidence region.

Theorem 4.2. *The total expected regret of* **MultiSBM** *(Algorithm 3) in the UBDB game is*

$$O\Big(H\alpha\ln T + H\alpha\Big(K\ln K + K\ln\ln T - \sum\nolimits_{x \neq x^*}\ln\Delta_x\Big)\Big),$$

assuming the policy of the SBMs defined in Line 1 is α-robust for $\alpha = \max(3, \ln(K)/\ln\ln(T))$. The robustness can be ensured by choosing the UCB policy (Algorithm 1) for the SBM with parameter α.

Note that achieving $(\alpha = 3)$-robustness requires implementing a variant of UCB with a slight modification of the confidence interval parameter in each SBM. Therefore,

---

[6]If this is not the case, our statements still hold, yet the proof becomes slightly more technical. As there is no real additional complication to the problem under this setting, we ignore this case.

if the horizon T is large enough so that $\ln \ln T > (\ln K)/3$, then the total regret is comparable to that of UCB in the standard MAB game.

The proof of the theorem is deferred to Appendix C. The main idea behind the proof is showing that a certain "positive feedback loop" emerges: if the expected regret incurred by the right arm at some time t is low, then there is a higher chance that x^* will be played as the left arm at time $t + 1$. Conversely, if any fixed arm (in particular, x^*) is played very often as the left arm, then the expected regret incurred by the right arm decreases rapidly.

5 A Heuristic Approach

In this section we describe a heuristic called **Sparring** for playing UBDB, which shows extremely good performance in our experiments. Unfortunately, as of yet we were unable to prove performance bounds that explain its empirical performance. **Sparring** uses two SBMs, corresponding to *left* and *right*. In each round the pair of arms is chosen according to the strategies of the two corresponding SBMs. The SBM corresponding to the chosen arm receives a feedback of 1 while the other receives 0. The formal algorithm is described in Algorithm 4.

The intuition for this idea comes from analysis of an adversarial version of UDBD, in which it can be easily shown that the resulting expected regret of **Sparring** is at most a constant times the regret of the two SBMs which emulate an algorithm for adversarial MAB. (We omit the exact discussion and analysis for the adversarial counterpart of UDBD in this extended abstract.) We conjecture that the regret of **Sparring** is asymptotically bounded by the combined regret of the algorithms hiding in the SBM's, with (possibly) a small overhead. Proving this conjecture is especially interesting for settings in which X is infinite and a MAB algorithm with polylogarithmic regret exists. Indeed, previous literature based on tournament elimination strategies does not apply to infinite X, and **Doubler** presented earlier is probably suboptimal due to the extra log-factor it incurs.

Proving the conjecture appears to be tricky due to the fact that the left (resp. right) SBM does not see a stochastic environment, because its feedback depends on non-stochastic choices made by the right (resp. left) SBM. Perhaps there exist bad settings where both strategies would be mutually 'stuck' in some sub-optimal state. We leave the analysis of this approach as an interesting problem for future research. Our experiments will nevertheless include **Sparring**.

6 Notes

Lower Bound: Our results contain upper bounds for the regret of the dueling bandit problem. We note that a matching lower bound, up to logarithmic terms can be shown via a simple reduction to the MAB problem. This reduction is the reverse of the others presented here: simulate a SBM by using a UBDB solver. It is an easy exercise to obtain such a reduction whose regret w.r.t. the MAB problem is at most twice the

Algorithm 4 (Sparring): Reduction to two SBMs.

1: $S_L, S_R \leftarrow$ two new SBMs over X
2: $\text{reset}(S_L), \text{reset}(S_R), t \leftarrow 1$
3: **while** true **do**
4:     $x_t \leftarrow \text{advance}(S_L); y_t \leftarrow \text{advance}(S_R)$
5:     play (x_t, y_t), observe choice $b_t \in \{0, 1\}$
6:     $\text{feedback}(S_L, \mathbf{1}_{b_t=0}); \text{feedback}(S_R, \mathbf{1}_{b_t=1})$
7:     $t \leftarrow t + 1$
8: **end while**

---

regret of the dueling bandit problem. It follows that the same lower bounds of the classic MAB problem apply to the UBDB problem.

Adversarial Setting: One may also consider an adversarial setting for the UBDB problem. Here, utilities of the arms that are assumed to be constant in the stochastic case are assumed to change each round in some arbitrary way. We do not elaborate on this setting due to space constraints but mention that (a) a lower bound of $\sqrt{KT}$ matching that of the MAB problem is valid in the UDBD setting, and (b) the **Sparring** algorithm, when using SBM solvers for the adversarial setting, can be shown to obtain the same regret bounds of said SBM solvers.

7 Experiments

We now present several experiments comparing our algorithms with baselines consisting of the state-of-the-art INTERLEAVED FILTER (**IF**) Yue et al. (2012) and BEAT THE MEAN BANDIT (**BTMB**) Yue & Joachims (2011). Our experiments are exhaustive, as we include scenarios for which no bounds were derived (e.g. nonlinear link functions), as well as the much more general scenario in which **BTMB** was analyzed Yue & Joachims (2011).

Henceforth, the set X of arms is $\{A, B, C, D, E, F\}$. For applications such as the interleaving search engines Chapelle et al. (2012), 6 arms is realistic. We considered 5 choices of the expected value function $\mu(\cdot)$ and 3 link functions[7][8].

linear	$\phi(x, y) = (1 + x - y)/2$				
natural	$\phi(x, y) = x/(x + y)$				
logit	$\phi(x, y) = (1 + \exp\{y - x\})^{-1}$				

Name	$\mu(A)$	$\mu(B)$	$\mu(C)$	$\mu(D)$	$\mu(E)$	$\mu(F)$
1good	0.8	0.2	0.2	0.2	0.2	0.2
2good	0.8	0.7	0.2	0.2	0.2	0.2
3good	0.8	0.7	0.7	0.2	0.2	0.2
arith	0.8	0.7	0.575	0.45	0.325	0.2
geom	0.8	0.7	0.512	0.374	0.274	0.2

---

[7]To be precise, the actual expected utility vector μ was a random permutation of the one given in the table. This was done to prevent initialization bias arising from the specific implementation of the algorithms.
[8]Note that in row 'arith', $\mu(2)..\mu(6)$ form an arithmetic progression, and in row 'geom' they form a geometric progression.

For each 15 combinations of arm values and link function we ran all 5 algorithms **IF**, **BTMB**, **MultiSBM**, **Doubler**, and **Sparring** with random inputs spanning a time horizon of up to 32000.

We also set out to test our algorithms in a scenario defined in Yue & Joachims (2011). We refer to this setting as YJ. Unlike our setting, where choice probabilities are derived from (random) latent utilities together with a link function, in YJ an underlying unknown fixed matrix (P_{xy}) is assumed, where P_{xy} is the probability of arm x chosen given the pair (x, y). The matrix satisfies very mild constraints. Following Yue & Joachims (2011), define $\epsilon_{xy} = (P_{xy} - P_{yx})/2$. The main constraint is, for some unknown total order $\succ$ over X, the imposition $x \succ y \iff \epsilon(x, y) > 0$. The optimal arm x^* is maximal in the total order. The regret incurred by playing the pair (x_t, y_t) at time t is $\frac{1}{2}(\epsilon_{x^* x_t} + \epsilon_{x^* y_t})$.

The **BTMB** algorithm Yue & Joachims (2011) proposed for YJ is, roughly speaking, a tournament elimination scheme, in which a working set of candidate arms is maintained. Arms are removed from the set whenever there is high certainty about their suboptimality. Although the YJ setting is more general than ours, our algorithms can be executed without any modification, giving rise to an interesting comparison with **BTMB**. For this comparison, we shall use the same matrix $(\epsilon_{xy})_{x,y \in X}$ as in Yue & Joachims (2011), which was empirically estimated from an operational search engine.

	A	B	C	D	E	F
A	0	0.05	0.05	0.04	0.11	0.11
B	-0.05	0	0.05	0.06	0.08	0.10
C	-0.05	-0.05	0	0.04	0.01	0.06
D	-0.04	-0.04	-0.04	0	0.04	0
E	-0.11	-0.08	-0.01	-0.04	0	0.01
F	-0.11	-0.10	-0.06	0	-0.01	0

(Note that $x^* = A \succ B \succ C \succ D \succ E \succ F$.)

Experiment Results and Analysis   Figure 1 contains the expected regrets of these described scenarios as a function of the $\log$ (to the base 2) of the time, averaged over 400 executions, with one standard deviation confidence bars. The experiments reveal some interesting results. First, the heuristic approach is superior to all others in all of the settings. Second, among the other algorithms, the top two are the algorithms **IF** and **MultiSBM**, where **MultiSBM** is superior in a wide variety of scenarios.

8   Future work

We dealt with choice in sets of size 2. What happens in cases where the player chooses from larger sets? We also analyzed only the linear choice function. See Appendix D for an extension of the results in Section 4 to other link functions.

Both algorithms **Doubler** and **MultiSBM** treated the left and right sides asymmetrically. This did not allow us to consider distinct expected valuation functions for the left and right positions. [9] Algorithm **Sparring** *is* symmetric, further motivating the question of proving its performance guarantees.

---

[9]Such a case is actually motivated in a setting where, say, the perceived user valuation of items appearing lower in the list are lower, giving rise to bias toward items appearing at the top.
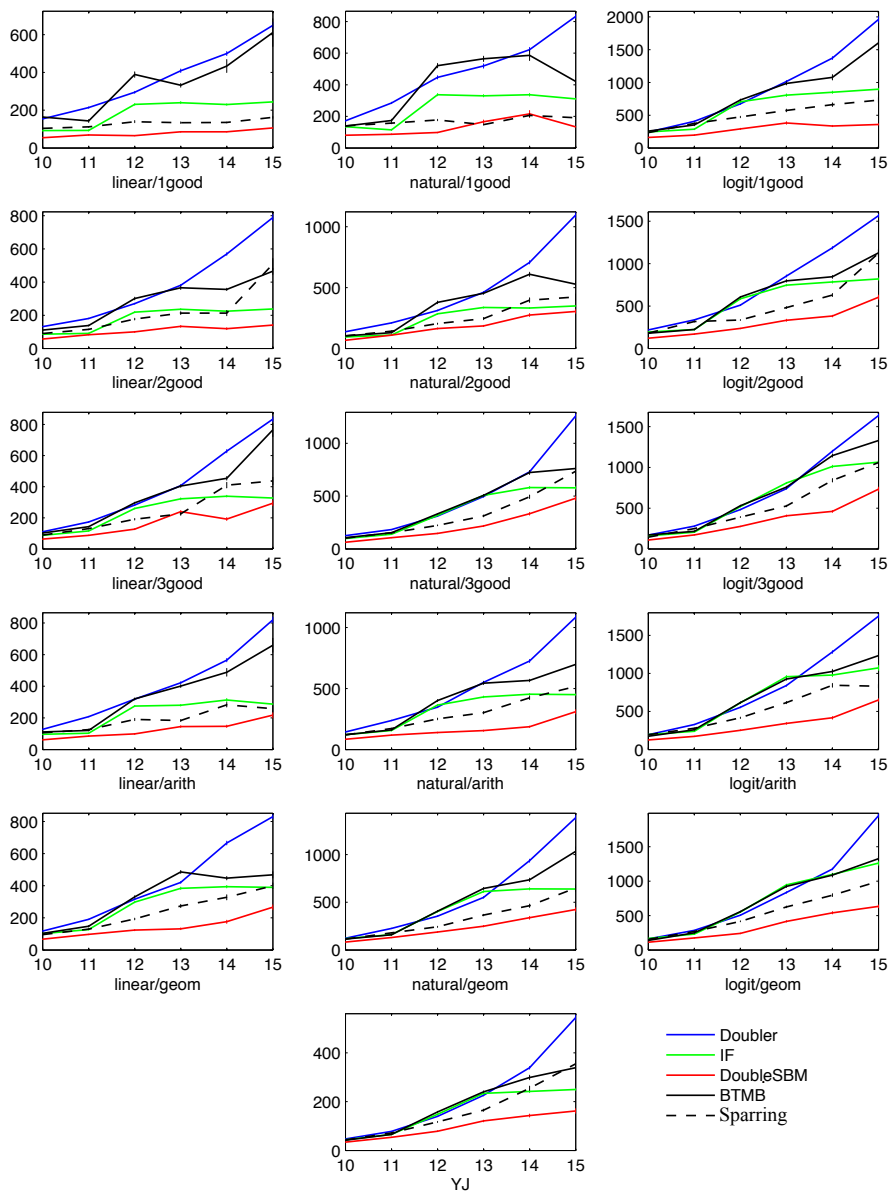
Figure 1: Expected regret plots, averaged over 400 runs for each of the 16 scenarios, and 5 algorithms. The x-axis is the log to the base 2 of the time, and the y-axis is the regret, averaged over 400 executions (with 1 standard deviation confidence bars).

Proving (or refuting) the conjecture in Section 5 regarding the regret of **Sparring** is an interesting open problem. Much like our proof idea for the guarantee of **MultiSBM**, there is clearly a positive feedback loop between the two SBM's in **Sparring**: the

13

more often the left (resp. right) arm is played optimally, the right (resp. left) arm would experience an environment which is closer to that of the standard MAB, and would hence incur expected regret approximately that of the SBM it implements.

Acknowledgments

References

Agarwal, Alekh, Dekel, Ofer, and Xiao, Lin. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pp. 28–40, 2010.

Ailon, Nir, Begleiter, Ron, and Ezra, Esther. Active learning using smooth relative regret approximations with applications. *Journal of Machine Learning Research - Proceedings Track*, 23:19.1–19.20, 2012.

Antos, András, Bartók, Gábor, Pál, Dávid, and Szepesvári, Csaba. Toward a classification of finite partial-monitoring games. *Theoretical Computer Science*, 2012.

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

Avner, Orly, Mannor, Shie, and Shamir, Ohad. Decoupling exploration and exploitation in multi-armed bandits. In *ICML*, 2012.

Bartók, Gábor, Zolghadr, Navid, and Szepesvári, Csaba. An adaptive algorithm for finite stochastic partial monitoring. *arXiv preprint arXiv:1206.6487*, 2012.

Chapelle, O., Joachims, T., Radlinski, F., and Yue, Yisong. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30(1):6:1–6:41, 2012.

Dani, Varsha, Hayes, Thomas P., and Kakade, Sham M. Stochastic linear optimization under bandit feedback. In *COLT*, pp. 355–366, 2008.

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

Freund, Yoav, Iyer, Raj D., Schapire, Robert E., and Singer, Yoram. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

Herbrich, R, Graepel, Thore, and Obermayer, Klaus. Large margin rank boundaries for ordinal regression. *Book chapter, Advances in Large Margin Classifiers*, 2000.

Joachims, T., Granka, L., Pan, Bing, Hembrooke, H., Radlinski, F., and Gay, G. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2), April 2007.

Karp, Richard M. and Kleinberg, Robert. Noisy binary search and its applications. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pp. 881–890, 2007.

Mannor, Shie and Shamir, Ohad. From bandits to experts: On the value of side-observations. In *NIPS*, pp. 684–692, 2011.

Piccolboni, Antonio and Schindelhauer, Christian. Discrete prediction games with arbitrary feedback and loss. In *Computational Learning Theory*, pp. 208–223. Springer, 2001.

Robbins, H. Some aspects of the sequential design of experiments. *Bulletin of the AMS*, 58:527–535, 1952.

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

Yue, Yisong and Joachims, T. Interactively optimizing information retrieval systems as a dueling bandits problem. In *International Conference on Machine Learning (ICML)*, pp. 151–159, 2009.

Yue, Yisong and Joachims, Thorsten. Beat the mean bandit. In *ICML*, pp. 241–248, 2011.

Yue, Yisong, Broder, Josef, Kleinberg, Robert, and Joachims, Thorsten. The k-armed dueling bandits problem. *J. Comput. Syst. Sci.*, 78(5):1538–1556, 2012.

A Robustness of the UCB algorithm

For completeness, we present a proof of robustness for the UCB algorithm, presented as Algorithm 1 below. Note that we did not make an effort to bound the constants in the proof. We start by presenting Chernoff's inequality providing a tail bound for estimations of variables contained in $[0, 1]$.

Lemma A.1. *Let $Y_1, \ldots, Y_t$ be i.i.d variables supported in $[0, 1]$. Then for any $\varepsilon > 0$ it holds that*

$$\Pr\left[\frac{1}{t}\sum_{i=1}^{t} Y_i - \mathbb{E}[Y_i] > \varepsilon\right] \leq e^{-2t\varepsilon^2}$$

Recall that in our setting, there are K arms, each with an expected reward. For convenience we assume the set of bandits X is the set $\{1, \ldots, K\}$ and further assume for the purpose of the analysis that arm 1 has the largest expected reward. We denote by Δ_i the difference between the reward of arm 1 and that of arm i.

Proof of Lemma 2.2. For convenience, define $\beta = \alpha + 2$ where α is the robustness parameter given as input to the algorithm. For $i > 1$, define

$$u_i(t) = 2\beta \ln(t)/\Delta_i^2$$

If at time t, arm i where $i > 1$ (i.e. i is suboptimal) was chosen, one of the following must be true

1. $\rho_i(t) < u_i(t)$

2. $\hat{\mu}_i > \mu_i + \sqrt{\frac{\beta \ln(t)}{2\rho_i(t)}}$

3. $\hat{\mu}_1 + \sqrt{\frac{\beta \ln(t)}{2\rho_1(t)}} < \mu_1$

Here, $\rho_i(t), \rho_1(t)$ denote the number of times arms i and 1 (the optimal arm) were pulled up to time t. Indeed, if all 3 are false we have

$$\hat{\mu}_1 + \sqrt{\frac{\beta \ln(t)}{2\rho_1(t)}} \geq \mu_1 = \mu_i + \Delta_i \geq$$

$$\mu_i + 2\sqrt{\frac{\beta \ln(t)}{2\rho_i(t)}} \geq \hat{\mu}_i + \sqrt{\frac{\beta \ln(t)}{2\rho_i(t)}}$$

and the i'th arm cannot be chosen. Hence, denoting $\rho_i(T)$ the number of times arm i is queried in a total budget of T queries, we have

$$\mathbb{E}[\rho_i(T) - u_i(T)] \leq \sum_{t=u_i(T)+1}^{T} \Pr[(2) \text{ or } (3)]$$

To bound the probability of event (2) occurring, we use Chernoff's inequality (Lemma A.1)

$$\Pr[(2)] \leq \Pr\left[\exists \rho_i \in [t] : \hat{\mu}_i > \mu_i + \sqrt{\frac{\beta \ln(t)}{2\rho_i}}\right] \leq$$

$$t \cdot t^{-\beta} = t^{1-\beta}.$$

The bound for event (3) is analogous. It follows that the probability of events (2) or (3) occurring is bounded by $2t^{1-\beta}$ and

$$\mathbb{E}[\rho_i(T) - u_i(T)] \leq \sum_{t=u_i(T)+1}^{T} 2t^{1-\beta} \leq$$

$$\frac{2}{\beta - 2} \left(2\beta \ln(T)\Delta_i^{-2}\right)^{2-\beta} \tag{A.1}$$

Proving the bound on the expected regret is now a matter of simple calculation

$$\mathbb{E}[R] = \sum_{i>1} \mathbb{E}[\Delta_i \cdot \rho_i(T)] \leq$$

$$\frac{2 \sum_i \Delta_i}{\beta - 2} + \sum_{i>1} 2\beta \ln(T)/\Delta_i \leq \frac{2K}{\beta - 2} + 2\beta H \ln(T)$$

We proceed to prove the high probability bounds on the number of pulls of a sub-optimal arm. Denote by $\rho_i^s(T)$ the number of times arm i was chosen starting from the time point $t \geq s$. Assuming $s \geq 2\beta \ln(T)\Delta_i^{-2}$, by the same arguments leading to equation A.1 we have

$$\mathbb{E}[\rho_i^s(T) - u_i(T)] \leq \frac{2}{\beta - 2} s^{2-\beta}$$

Assume that arm i was chosen at least s times for some

$$s \geq \frac{4(\beta + 2) \ln(T)}{\Delta_i^2}$$

it follows that $\rho_i^{s-u_i(T)-1}(T) \geq u_i(T) + 1$. The probability of this happening is bounded by Markov's inequality by

$$\Pr[\rho_i(T) \geq s] \leq \Pr\left[\rho_i^{s-u_i(T)-1}(T) - u_i(T) \geq 1\right] \leq$$

$$\mathbb{E}\left[\rho_i^{s-u_i(T)-1}(T) - u_i(T)\right] \leq$$

$$\frac{2}{\beta - 2}(s - u_i(T) - 1)^{2-\beta} \leq \frac{2}{\beta - 2}\left(\frac{s}{2}\right)^{2-\beta}$$

The last inequality holds since

$$s \geq \frac{4(\beta + 2) \ln(T)}{\Delta_i^2} \geq 2 + \frac{4\beta \ln(T)}{\Delta_i^2} = 2u_i(T) + 2$$

$\square$

B Proof of Theorem 3.1

Let $B(T)$ denote the supremum of the expected regret of the SBM S (defined in line1 of Algorithm 2) after T steps, over all possible utility distributions of the arm set X.

Fix a phase i in the algorithm. The length T_i of the phase is exactly 2^i. For all time steps t inside the phase, the left bandit x_t is drawn from some fixed distribution. Let μ' denote the common expectation $\mathbb{E}[u_t] = \mathbb{E}_{x_t}[u_t|x_t]$ of the reward of the left arm in all steps t in the phase. Now, the SBM S (defined in Line 1) is playing a standard MAB

game over the set X with binary rewards. Let b_t denote the binary reward in the t'th step (within the phase). By construction,

$$\mathbb{E}[b_t|v_t, u_t] = \frac{v_t - u_t + 1}{2} \in [0, 1] .$$ (B.1)

By conditional expectation, for all $y \in X$,

$$\mathbb{E}[b_t|y_t = y] = \frac{\mu(y) - \mu' + 1}{2} \in [0, 1] .$$ (B.2)

Note that the arm with highest expected reward is $y = x^*$. By the definition of the bound function $B(T)$, the total expected regret (in the traditional MAB sense) of the SBM S in the phase is at most $B(T_i) = B(2^i)$. This means, that

$$\mathbb{E}\left[\sum_t \left(b_t - \frac{\mu(x^*) - \mu' + 1}{2}\right)\right] \le B(2^i) ,$$

where the summation runs over t in the phase. But this clearly means, using (B.2), that

$$\mathbb{E}\left[\sum_t \frac{\mu(y_t) - \mu(x^*)}{2}\right] \le B(2^i) .$$

But notice that $\mathbb{E}[v_t] = \mathbb{E}_{y_t}\mathbb{E}[v_t|y_t] = \mathbb{E}[\mu(y_t)]$. Hence,

$$\mathbb{E}\left[\sum_t \frac{v_t - \mu(x^*)}{2}\right] \le B(2^i) .$$

In words, this says that the expected contribution of the *right arm* to the regret (in the UDBD game) in phase i is at most $B(2^i)$. It remains to bound the expected contribution to the regret of the left bandit in phase i, which is drawn by a distribution which assigns to all $x \in X$ a probability proportional to the frequency of x played as the *right arm* in the *previous phase*.[10] By the principle of conditional expectation, and due to the linearity of the link function, the expected regret incurred by x_t (in each step in the phase) is *exactly* the average expected regret contributed by the right bandit in phase $i - 1$, and hence at most $B(2^{i-1})/2^{i-1}$. This means that the total expected regret incurred by the left bandit in phase i is bounded by $2^i(B(2^{i-1})/2^{i-1}) = 2B(2^{i-1})$. Concluding, for a time horizon of T uniquely decomposable as $2 + 4 + 8 + \cdots + 2^k + Z$ for some integers $k \ge 1$ and $0 \le Z \le 2^{k+1}$-1, the total expected regret is given by the following function of T:

$$1/2 + 3B(2) + 3B(4) + \cdots + 3B(2^k) + B(Z) .$$ (B.3)

The theorem claim is now obtained by simple analysis of (B.3).

---

[10]If X is infinite, to be precise we need to say that the distribution is also supported on the set of arms played on the right side in the previous phase.

C Proof of Theorem 4.2

To follow the proof, it is important to understand that in **MultiSBM** (Algorithm 3), exactly one SBM is advanced at each step in Line 6. This means that the internal timer of each SBM may be (and usually is) strictly behind the iteration counter of the algorithm, which is measured by the variable t. Denote by $\rho_x(t)$ the total number of times S_x was advanced after t iterations of the algorithm, for all $x \in X$.

We now assume that all coin tosses are fixed (obliviously) in advance. This allows us to discuss the regret of the SBM S_x (line 1) after T' *internal* steps even if in practice the value t for which $\rho_x(t) = T'$ might be much larger than the total number of arm pulls T, and in fact, may not even exist.

Notice that internally, S_x sees a world in which the reward is binary, and the expected reward for bandit $y \in X$ is exactly $(\mu(y) - \mu(x) + 1)/2$ at each internal step. This is because when S_x is advanced, the left bandit (in the UBDB game) is identically x. It follows that in all SBMs, the suboptimalities are the same and are $\Delta_y/2$ for arm y.

For $x \in X$ and integer $T' > 0$, let

$$R_x(T') = \frac{1}{2} \sum_{t:\rho_x(t)\leq T',x_t=x} \Delta_{y_t}$$

In words, this is the contribution of the right bandit choices to the UBDB regret at all times t for which the left bandit is chosen as x, and S_x's internal counter has not surpassed T'. The expression $R_x(T')$, by the last discussion, also measures the expected internal regret seen by S_x after T' internal steps. Similarly, we define

$$R_{xy}(T') = \#\{t : \rho_x(t) \leq T', x_t = x, y_t = y\}\Delta_y/2$$

This measures a part of $R_x(T')$ for which the right bandit is y. We start with an observation expressing the regret of the entire process as a function of the different R_{xy}'s. It will be useful to define $\rho_{xy}(T') = \#\{t : \rho_x(t) \leq T', x_t = x, y_t = y\}$, so that $R_{xy}(T') = \rho_{xy}(T')\Delta_y/2$.

Observation C.1. *For any $T \geq 1$, the total regret $R(T)$ of* **MultiSBM** *after T steps satisfies* $\left|R(T) - 2 \sum_{x\in X} \sum_{y\in X} R_{xy}(\rho_x(T))\right| \leq 0.5.$

We conclude that in order to bound the expected regret $R(T)$ it suffices to bound the expressions $\mathbb{E}[R_{xy}(\rho_x(T))]$. By using the upper bound of $\rho_x(T) \leq T$, we get the trivial bound for $\mathbb{E}[R(T)]$ of K times the expected regret of a single machine. The main insight is to exploit the fact that typically, $\rho_x(T)$ is order of $\ln T$ for suboptimal x. We begin with the observation that for any fixed $x, y \in X$ (x suboptimal), $s \geq 8\alpha$,

$$\begin{aligned}
&\Pr[R_{xy}(T) \geq (s\ln T)/\Delta_y] \\
&= \Pr[R_{xy}(T) \geq ((s/2)\ln T)/(\Delta_y/2)] \\
&= \Pr[\rho_{xy}(T) \geq ((s/2)\ln T)/(\Delta_y/2)^2] \\
&\leq \left((s/4)\ln T)/(\Delta_y/2)^2\right)^{-\alpha} \leq (s\ln T)^{-\alpha} \quad\quad \text{(C.1)}
\end{aligned}$$

This is immediate from the α-robustness of the SBM and the fact we choose $\alpha > 2$. For the same assumption on s and x, y and using the union bound,

$$\Pr\left[\exists p \in \{0, \ldots, \lceil \ln \ln T \rceil\} : R_{xy}\left(e^{e^p}\right) \geq s \cdot p/\Delta_y\right]$$
$$\leq 2s^{-\alpha} \qquad (C.2)$$

We now bound the quantity $\rho_x(T)$ for any nonoptimal fixed x. Using the (trivial) fact that all $z \in X$ satisfy $\rho_z(T) \leq T$, together with the fact that SBM S_x is advanced in each iteration only if x was the right bandit in the previous one, we have that for all suboptimal x,

$$\Pr[\rho_x(T) \geq (sK \ln T)/\Delta_x^2]$$
$$\leq \sum_{z \in X} \Pr\left[R_{zx}(T) \geq (s \ln T)/\Delta_x\right] \leq K/(s \ln T)^\alpha, \qquad (C.3)$$

where the rightmost inequality is by union bound and (C.1). Fix some $x, y \in X$ (x suboptimal). The last two inequalities give rise to a random variable Z defined as the minimal scalar for which we have

$$\forall T' \in [e, e^e, e^{e^2}, \ldots, e^{e^{\lceil \ln \ln(T) \rceil}}],$$

$$\rho_x(T) \leq (ZK \ln T)/\Delta_x^2, \quad R_{xy}(T') \leq (Z \ln T')/\Delta_y$$

By (C.2)-(C.3) we have that for all $s \geq 8\alpha$, $Pr[Z \geq s] \leq 2s^{-\alpha} + K(s \ln T)^{-\alpha}$. Also, conditioned on the event that $\{Z \leq s\}$ we have that $R_{xy}(\rho_x(T)) \leq R_{xy}^s :=$ $s \cdot e \cdot \ln((sK \ln T)/\Delta_x^2)/\Delta_y$, which is $O\left(s\Delta_y^{-1}\left(\ln \ln T + \ln K + \ln s + \ln(1/\Delta_x)\right)\right)$. Combining, $\mathbb{E}[R_{xy}(\rho_x(T))]$ is bounded above by:

$$R_{xy}^{8\alpha-1} + \sum_{i=0}^\infty R_{xy}^{8\alpha+i}(2(8\alpha + i)^{-\alpha} + K((8\alpha + i) \ln T)^{-\alpha}).$$

For $\alpha = \max\{3, 2 + (\ln K)/\ln \ln T\}$, it is easy to verify that the last expression converges to $O(R_{xy}^{8\alpha})$, hence

$$\mathbb{E}[R_{xy}(\rho_x(T))] = O\left(\alpha\Delta_y^{-1}\left(\ln \ln T + \ln K + \ln(1/\Delta_x)\right)\right).$$

Concluding, the total expected regret $\mathbb{E}[R]$ is at most $0.5 + \mathbb{E}[R_{x^*} + \sum_{x,y \in X \setminus \{x^*\}} R_{xy}]$, clearly proving the theorem.

D Extension to more General Models

Assume the setting of Section 4. In this section we assume for simplicity that for any t and any choice of x_t, y_t, the utilities are deterministically $u_t = \mu(x_t), v_t = \mu(y_t)$. In Yue & Joachims (2011), the dueling bandit problem is presented where a more relaxed assumption is made on the probabilities of the outcomes of duels. Each pair of arm x, y is assigned a parameter $\Delta(x, y)$ such that the probability of x being chosen when dueling with y is $0.5 + \Delta(x, y)$. It is assumed that there exists some order $\succ$ over the arms and the Δ's hold two properties.

- *(Relaxed) Stochastic Transitivity*: For some $\gamma \geq 1$ and any pair $x^* \succ x \succ y$ we have $\gamma\Delta(x^*, y) \geq \max\{\Delta(x^*, x), \Delta(x, y)\}$.

- *(Relaxed) Stochastic Triangle Inequality*: For some $\gamma \geq 1$ and any pair $x^* \succ x \succ y$ we have $\gamma\Delta(x^*, y) \leq \Delta(x^*, x) + \Delta(x, y)$.

We have analyzed **MultiSBM** (Algorithm 3) under the assumption that $\Delta(x, y) = (\mu(x) - \mu(y))/2$. It can be easliy verified that our proof holds for arbitrary Δ's under the following assumption:

- *(Relaxed) Extended Stochastic Triangle Inequality*. For some $\gamma \geq 1$, and any pair x, y (where it does not necessarily hold that $x \succ y$) it holds that $\gamma\Delta(x^*, y) \leq \Delta(x^*, x) + \Delta(x, y)$.

This property is clearly held for $\Delta(x, y) = (\mu(x) - \mu(y))/2$. However, it holds for a wider family of Δ's. For example, it holds for $\Delta(x, y) = \mu(x)/(\mu(x) + \mu(y))$, assuming all μ's are in the region $[1/\gamma, 1]$. The effect of γ to the regret is given in the following theorem:

Theorem D.1. *Assume the probability for the outcome of a duel is defined according to $\Delta(x, y)$, where Δ has the Relaxed Extended Stochastic Triangle Inequality with parameter γ. The total expected regret of* **MultiSBM** *in the UBDB game is asymptotic to*

$$\gamma H\alpha \left(K\ln(K) + K\ln\ln(T) + \sum_{x \in X \setminus \{x^*\}} \ln(1/\Delta_x) \right) +$$

$$\ln(T)H\alpha$$

assuming the invoked MAB policy is α-robust for $\alpha = \max(3, \ln(K)/\ln\ln(T))$.

Notice that γ does not enter the summand of $\ln(T)$, meaning that for large values of T, the regret is unaffected by γ. We defer the proof of the theorem to the full version of the paper.

E Proof of Observation 2.1

By definition,

$$\mathbb{E}[R_t^{\text{choice}}|(x_t, y_t)] = \mu(x^*) - \mathbb{E}[U_t^{\text{choice}}|(x_t, y_t)] .$$

But now note that by the definition of the link function and of U_t^{choice},

$$\mathbb{E}[U_t^{\text{choice}}|(x_t, y_t)] = \phi(u_t, v_t)u_t + \phi(v_t, u_t)v_t \geq \frac{u_t + v_t}{2}$$

where we used the assumption that for $u > v$, $\phi(u, v) > 1/2$. Now notice that the expression on the right is exactly $\mathbb{E}[U^{\text{av}}|(x_t, y_t)]$. Hence,

$$\mathbb{E}[R_t^{\text{choice}}|(x_t, y_t)] \leq \mu(x^*) - \mathbb{E}[U_t^{\text{av}}|(x_t, y_t)] = \mathbb{E}[R_t^{\text{av}}] .$$