Multi-dueling Bandits with Dependent Arms

Yanan Sui
Caltech
Pasadena, CA 91125
ysui@caltech.edu

Vincent Zhuang
Caltech
Pasadena, CA 91125
vzhuang@caltech.edu

Joel W. Burdick
Caltech
Pasadena, CA 91125
jwb@robotics.caltech.edu

Yisong Yue
Caltech
Pasadena, CA 91125
yyue@caltech.edu

Abstract

The dueling bandits problem is an online learning framework for learning from pairwise preference feedback, and is particularly well-suited for modeling settings that elicit subjective or implicit human feedback. In this paper, we study the problem of *multi-dueling bandits with dependent arms*, which extends the original dueling bandits setting by simultaneously dueling multiple arms as well as modeling dependencies between arms. These extensions capture key characteristics found in many real-world applications, and allow for the opportunity to develop significantly more efficient algorithms than were possible in the original setting. We propose the SELFSPARRING algorithm, which reduces the multi-dueling bandits problem to a conventional bandit setting that can be solved using a stochastic bandit algorithm such as Thompson Sampling, and can naturally model dependencies using a Gaussian process prior. We present a no-regret analysis for multi-dueling setting, and demonstrate the effectiveness of our algorithm empirically on a wide range of simulation settings.

1 INTRODUCTION

In many online learning settings, particularly those that involve human feedback, reliable feedback is often limited to pairwise preferences (e.g., "is A better than B?"). Examples include implicit or subjective feedback for information retrieval and various recommender systems (Chapelle et al., 2012; Sui & Burdick, 2014). This setup motivates the dueling bandits problem (Yue et al., 2012), which formalizes the problem of online regret minimization via preference feedback.

The original dueling bandits setting ignores many real world considerations. For instance, in personalized clinical recommendation settings (Sui & Burdick, 2014), it is often more practical for subjects to provide preference feedback on several actions (or treatments) simultaneously rather than just two. Furthermore, the action space can be very large, possibly infinite, but often has a low-dimensional dependency structure.

In this paper, we address both of these challenges in a unified framework, which we call *multi-dueling bandits with dependent arms*. We extend the original dueling bandits problem by simultaneously dueling multiple arms as well as modeling dependencies between arms using a kernel. Explicitly formalizing these real-world characteristics provides an opportunity to develop principled algorithms that are much more efficient than algorithms designed for the original setting. For instance, most dueling bandits algorithms suffer regret that scales linearly with the number of arms, which is not practical when the number of arms is very large or infinite.

For this setting, we propose the SELFSPARRING algorithm, inspired by the Sparring algorithm from Ailon et al. (2014), which algorithmically reduces the multi-dueling bandits problem into a conventional muilti-armed bandit problem that can be solved using a stochastic bandit algorithm such as Thompson Sampling (Chapelle & Li, 2011; Russo & Van Roy, 2014). Our approach can naturally incorporate dependencies using a Gaussian process prior with an appropriate kernel.

While there have been some prior work on multi-dueling (Brost et al., 2016) and learning from pairwise preferences over kernels (Gonzalez et al., 2016), to the best of our knowledge, our approach is the first to address to both in a unified framework. We are also the first to provide a regret analysis of the multi-dueling setting. We further demonstrate the effectiveness of our approach over conventional dueling bandits approaches in a wide range of simulation experiments.

2 BACKGROUND

2.1 Dueling Bandits

The original dueling bandits problem is a sequential optimization problem with relative feedback. Let $\mathcal{B} = \{b_1, \ldots, b_K\}$ be the set of K bandits (or arms). At each iteration, the algorithm duels or compares a single pair of arms b_i, b_j from the set of K arms (b_i and b_j can be identical). The outcome of each duel between b_i and b_j is an independent sample of a Bernoulli random variable. We define the probability that arm b_i beats b_j as:

$$P(b_i \succ b_j) = \phi(b_i, b_j) + 1/2,$$

where $\phi(b_i, b_j) \in [-1/2, 1/2]$ denotes the stochastic preference between b_i and b_j, thus $b_i \succ b_j \Leftrightarrow \phi(b_i, b_j) > 0$. We assume there is a total ordering, and WLOG that $b_i \succ b_j \Leftrightarrow i < j$.

The setting proceeds in a sequence of iterations or rounds. At each iteration t, the decision maker must choose a pair of bandits $b_t^{(1)}$ and $b_t^{(2)}$ to compare, and observes the outcome of that comparison. The quality of the decision making is then quantified using a notion of cumulative regret of T iterations:

$$R_T = \sum_{t=1}^{T} \left[\phi(b_1, b_t^{(1)}) + \phi(b_1, b_t^{(2)}) \right]. \tag{1}$$

When the algorithm has converged to the best arm b_1, then it can simply duel b_1 against itself, thus incurring no additional regret. In the recommender systems setting, one can interpret (1) as a measure of how much the user(s) would have preferred the best bandit over the the ones presented by the algorithm.

To date, there have been several algorithms proposed for the stochastic dueling bandits problem, including Interleaved Filter (Yue et al., 2012), Beat the Mean (Yue & Joachims, 2011), SAVAGE (Urvoy et al., 2013), RUCB (Zoghi et al., 2014, 2015b), Sparring (Ailon et al., 2014; Dudík et al., 2015), RMED (Komiyama et al., 2015), and DTS (Wu & Liu, 2016). Our proposed approach, SELFSPARRING, is inspired by Sparring, which along with RUCB-style algorithms are the best performing methods. In contrast to Sparring, which has no theoretical guarantees, we provide no-regret guarantees for SELFSPARRING, and demonstrate significantly better performance in the multi-dueling setting.

Previous work on extending the original dueling bandits setting have been largely restricted to settings that duel a single pair of arms at a time. These include continuous-armed convex dueling bandits (Yue & Joachims, 2009), contextual dueling bandits which also introduces the von Neumann winner solution concept (Dudík et al., 2015), sparse dueling bandits that focuses on the Borda winner solution concept (Jamieson et al., 2015), Copeland dueling bandits that focuses on the Copeland winner solution concept (Zoghi et al., 2015a), and adversarial dueling bandits (Gajane et al., 2015). In contrast, our work studies the complementary directions of how to formalize multiple duels simultaneously, as well as how to reduce the dimensionality of modeling the action space using a low-dimensional similarity kernel.

Recently, there have been increasing interest in studying personalization settings that simultaneously elicit multiple pairwise comparisons. Example settings include information retrieval (Hofmann et al., 2011; Schuth et al., 2014, 2016) and clinical treatment (Sui & Burdick, 2014). There have also been some previous work on multi-dueling bandits settings (Brost et al., 2016; Sui & Burdick, 2014; Schuth et al., 2016), however the previous approaches are limited in their scope and lack rigorous theoretical guarantees. In contrast, our approach can handle a wide range of multi-dueling mechanisms, has near-optimal regret guarantees, and can be easily composed with kernels to model dependent arms.

2.2 Multi-armed Bandits

Our proposed algorithm, SELFSPARRING, utilizes a multi-armed bandit (MAB) algorithm as a subroutine, and so we provide here a brief formal description of the conventional MAB problem for completeness. The stochastic MAB problem (Robbins, 1952) refers to an iterative decision making problem where the algorithm repeatedly chooses among K actions (or bandits or arms). In contrast to the dueling bandits setting, where the feedback is relative between two arms, here, we receive an absolute reward that depends on the arm selected. We assume WLOG that every reward is bounded between $[0, 1]$.[1] The goal then is to minimize the cumulative regret compared to the best arm:

$$R_T^{\text{MAB}} = \sum_{t=1}^{T} \left[\mu^1 - \mu(b_t) \right], \tag{2}$$

where b_t denotes the arm chosen at time t, $\mu(b)$ denotes the expected reward of arm b, and $\mu^1 = \operatorname{argmax}_b \mu(b)$. Popular algorithms for the stochastic setting include UCB (upper confidence bound) algorithms (Auer et al., 2002a), and Thompson Sampling (Chapelle & Li, 2011; Russo & Van Roy, 2014).

In the adversarial setting, the rewards are chosen in an adversarial fashion, rather than sampled independently

[1] So long as the rewards are bounded, one can shift and rescale them to fit within $[0, 1]$.

Algorithm 1 Thompson Sampling for Bernoulli Bandits

1: For each arm $i = 1, 2, \cdots, K$, set $S_i = 0$, $F_i = 0$.
2: **for** $t = 1, 2, \ldots$ **do**
3: For each arm $i = 1, 2, \cdots, K$, sample θ_i from $Beta(S_i + 1, F_i + 1)$
4: Play arm $i(t) := \text{argmax}_i \, \theta_i(t)$, observe reward r_t
5: $S_i \leftarrow S_i + r_t$, $F_i \leftarrow F_i + 1 - r_t$
6: **end for**

from some underlying distribution. In this case, regret (2) is rephrased as the difference in the sum of rewards. The predominant algorithm for the adversarial setting is EXP3 (Auer et al., 2002b).

2.3 Thompson Sampling

The specific MAB algorithm used by our SELFSPARRING approach is Thompson Sampling. Thompson Sampling is a stochastic algorithm that maintains a distribution over the arms, and chooses arms by sampling (Chapelle & Li, 2011). This distribution is updated using reward feedback. The entropy of the distribution thus corresponds to uncertainty regarding which is the best arm, and flatter distributions lead to more exploration.

Consider the Bernoulli bandits setting where observed rewards are either 1 (win) or 0 (loss). Let S_i and F_i denote the historical number of wins and losses of arm i, and let D_t denote the set of all parameters at round t:

$$D_t = \{S_1, \cdots, S_K; F_1, \cdots, F_K\}_t.$$

For brevity, we often represent D_t by D, since only the current iteration matters at run-time. The sampling process of Beta-Bernoulli Thompson Sampling given D is:

- For each arm i, sample $\theta_i \sim Beta(S_i + 1, F_i + 1)$.
- Choose the arm with maximal θ_i.

In other words, we model the average utility of each arm using a Beta prior, and rewards for arm i as Bernoulli distributed according to latent mean utility θ_i. As we observe more rewards, we can compute the posterior, which is also Beta distributed by conjugation between Beta and Bernoulli. The sampling process above can be shown to be sampling for the following distribution:

$$P(i|D) = P(i = \underset{b}{\text{argmax}} \, \theta_b | D). \tag{3}$$

Thus, any arm i is chosen with probability that it has maximal reward under the Beta posterior. Algorithm 1 describes the Beta-Bernoulli Thompson Sampling algorithm, which we use as a subroutine for our approach. Thompson Sampling enjoys near-optimal regret guarantees in the stochastic MAB setting, as given by the lemma below (which is a direct consequence of main theorems in Agrawal & Goyal (2012); Kaufmann et al. (2012)).

Lemma 1. *For the K-armed stochastic MAB problem, Thompson Sampling has expected regret:* $\mathbb{E}[R_T^{MAB}] = \mathcal{O}\left(\frac{K}{\Delta} \ln T\right)$*, where* Δ *is the difference between expected rewards of the best two arms.*

2.4 Gaussian Processes & Kernels

Normally, when one observes measurements about one arm (in both dueling bandits and conventional multi-armed bandits), one cannot use that measurement to infer anything about other arms – i.e., the arms are independent. This limitation necessarily implies that regret scales linearly w.r.t. the number of arms K, since each arm must be explored at least once to collect at least one measurement about it. We will use Gaussian processes and kernels to model dependencies between arms.

For simplicity, we present Gaussian processes in the context of multi-armed bandits. We will describe how to apply them to multi-dueling bandits in Section 3 A Gaussian process (GP) is a probability measure over functions such that any linear restriction is multivariate Gaussian. A GP is fully determined by its mean and a positive definite covariance operator, also known as a kernel. A $GP(\mu(b), k(b, b'))$ is a probability distribution across a class of "smooth" functions, which is parameterized by a kernel function $k(b, b')$ that characterizes the smoothness of f. One can think of f has corresponding to the reward function in the standard MAB setting.

We assume WLOG that $\mu(b) = 0$, and that our observations are perturbed by i.i.d. Gaussian noise, i.e., for samples at points $A_T = [b_1 \ldots b_T]$, we have $y_t = f(b_t) + n_t$ where $n_t \sim \mathcal{N}(0, \sigma^2)$ (we will relax this later). The posterior over f is then also Gaussian with mean $\mu_T(b)$, covariance $k_T(b, b')$ and variance $\sigma_T^2(b)$ that satisfy:

$$\begin{aligned} \mu_T(b) &= k_T(b)^T (\mathcal{K}_T + \sigma^2 I)^{-1} y_T \\ k_T(b, b') &= k(b, b') - k_T(x)^T (\mathcal{K}_T + \sigma^2 I)^{-1} k_T(b') \\ \sigma_T^2(b) &= k_T(b, b), \end{aligned}$$

where $k_T(b) = [k(b_1, b) \ldots k(b_T, b)]^T$ and $\mathcal{K}_T$ is the positive definite kernel matrix $[k(x, x')]_b x, b' \in A_T]$.

Posterior inference updates the mean reward estimates for all the arms that share dependencies (as specified by the kernel) with the arms selected for measurement. Thus one can show that MAB algorithms using Gaussian processes have regret that scale linearly w.r.t. the dimensionality of the kernel rather than the number of arms (which can now be infinite) (Srinivas et al., 2010).

3 MULTI-DUELING BANDITS

We now formalize the multi-dueling bandits problem. We inherit all notation from original dueling bandits setting (Section 2.1). The key difference is that the algorithm now selects a (multi-)set S_t of arms at each iteration t, and observes outcomes of duels between some pairs of arms in S_t. For example, in information retrieval this can be implemented via multi-leaving (Schuth et al., 2014) the ranked lists of the subset, S_t, of rankers and then inferring the relative quality of the lists (and the corresponding rankers) from user feedback.

In general, we assume the number of arms being dueled at each iteration is some fixed constant $m = |S_t|$. When $m = 2$, the problem reduces to the original dueling bandits setting. Extending the regret formulation from the original setting (1), we can write the regret as:

$$R_T = \sum_{t=1}^{T} \sum_{b \in S_t} \phi(b_1, b). \tag{4}$$

The goal then is to select subsets of arms S_t so that the cumulative regret (4) is minimized. Intuitively, all arms have to be selected a small number of times in order to be explored, but the goal of the algorithm is to minimize the number of times when suboptimal arms are selected. When the algorithm has converged to the best arm b_1, then it can simply choose S_t to only contain b_1, thus incurring no additional regret.

Our setting differs from Brost et al. (2016) in two ways. First, we play a fixed, rather than variable, number of arms at each iteration. Furthermore, we focus on total regret, rather than the instantaneous average regret in a single iteration; in many applications (e.g., Sui & Burdick (2014)), playing each arm incurs its own regret .

Feedback Mechanisms. Simultaneously dueling multiple arms opens up multiple options for collecting feedback. For example, in some applications it may be viable to collect all pairwise feedback for all chosen arms S_t. In other applications, it is more realistic to only observe the "winner" of S_t, in which we observe feedback that one $b \in S_t$ wins against all other arms in S_t, but nothing about pairwise preferences between the other arms.

Approximate Linearity. One assumption that we leverage in developing our approach is *approximate linearity*, which fully generalizes the linear utility-based dueling bandits setting studied in Ailon et al. (2014). For any triplet of bandits $b_i \succ b_j \succ b_k$ and some constant $\gamma > 0$:

$$\phi(b_i, b_k) - \phi(b_j, b_k) \geq \gamma \phi(b_i, b_j). \tag{5}$$

To understand Approximate Linearity, consider the special case when the preference function follows the form

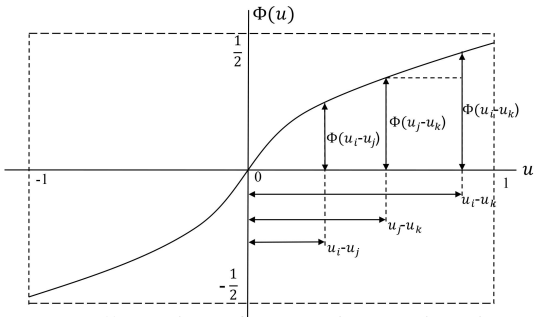


Figure 1: Illustration of Approximate Linearity. The curve represents $\Phi(\cdot)$ with support on $[-1, 1]$. Monotonicity guarantees Approximate Linearity for some γ.

$\phi(b_i, b_j) = \Phi(u_i - u_j)$, where u_i is a bounded utility measure of b_i. Approximate linearity of $\phi(\cdot, \cdot)$ is equivalent to having $\Phi(\cdot)$ be not far from some linear function on its bounded support (see Figure 1), and is satisfied by any continuous monotonic increasing function. When Φ is linear, then our setting reduces to the utility-based dueling bandits setting of Ailon et al. (2014).[2]

4 ALGORITHMS & RESULTS

We start with a high-level description of our general framework, called SELFSPARRING, which is inspired by the Sparring algorithm from Ailon et al. (2014). The high-level strategy is to reduce the multi-dueling bandits problem to a multi-armed bandit (MAB) problem that can be solved using a MAB algorithm, and ideally lift existing MAB guarantees to the multi-dueling setting.

Algorithm 2 describes the SELFSPARRING approach. SELFSPARRING uses a stochastic MAB algorithm such as Thompson sampling as a subroutine to independently sample the set of m arms, S_t to duel. The distribution of S_t is generally not degenerate (e.g., all the same arm) unless the algorithm has converged. In contrast, the Sparring algorithm uses m MAB algorithms to control the choice of the each arm, which essentially reduces the conventional dueling bandits problem to two multi-armed bandit problems "sparring" against each other.

SELFSPARRING takes as input S the total set of arms, m the number of arms to be dueled at each iteration, and η the learning rate for posterior updates. S can be a finite set of K arms for independent setting, or a continuous action space of arms for kernelized setting. A prior

[2]Compared to the assumptions of Yue et al. (2012), Approximate Linearity is a stricter requirement than strong stochastic transitivity, and is a complementary requirement to stochastic triangle inequality. In particular, stochastic triangle inequality requires that the curve in Figure 1 exhibits diminishing returns in the top-right quadrant (i.e., is sub-linear), whereas Approximate Linearity requires that the curve be not too far from linear.

Algorithm 2 SELFSPARRING

input arms $1, \ldots, K$ in space S, m the number of arms drawn at each iteration, η the learning rate

1: Set prior D_0 over S
2: **for** $t = 1, 2, \ldots$ **do**
3: **for** $j = 1, \ldots, m$ **do**
4: select arm $i_j(t)$ using D_{t-1}
5: **end for**
6: Play m arms $\{i_j(t)\}_j$ and observe $m \times m$ pairwise feedback matrix $R = \{r_{ij} \in \{0, 1, \emptyset\}\}_{m \times m}$
7: update D_{t-1} using R to obtain D_t
8: **end for**

distribution D_0 is used to initialize the sampling process over S. In the t-th iteration, SELFSPARRING selects m arms by sampling over the distribution D_{t-1} as shown in line 2 of Algorithm 2. The preference feedback can be any type of comparisons ranging from full comparison over the m arms (a full matrix for R, aka 'all pairs") to single comparison of one pair (just two valid entries in R). The posterior distribution over arms D_t then gets updated by R and the prior D_{t-1}.

We specialize SELFSPARRING in two ways. The first, INDSELFSPARRING (Algorithm 3), is the independent-armed version of SELFSPARRING. The second, KERNELSELFSPARRING (Algorithm 4), uses Gaussian processes to make predictions about preference function f based on noisy evaluations over comparisons. We emphasize here that SELFSPARRING is very modular approach, and is thus easy to implement and extend.

4.1 Independent Arms Case

INDSELFSPARRING (Algorithm 3) instantiates SELFSPARRING using Beta-Bernoulli Thompson sampling. The posterior Beta distributions D_t over the arms are updated by the preference feedback within the iteration and the prior Beta distributions D_{t-1}.

We present a no-regret guarantee of INDSELFSPARRING in Theorem 2 below. We now provide a high-level outline of the main components leading to the result. Detail proofs are deferred to the supplementary material.

Our first step is to prove that INDSELFSPARRING is asymptotically consistent, i.e., it is guaranteed (with high probability) to converge to the best bandit. In order to guarantee consistency, we first show that all arms are sampled infinitely often in the limit.

Lemma 2. *Running* INDSELFSPARRING *with infinite time horizon will sample each arm infinitely often.*

In other words, Thompson sampling style algorithms do not eliminate any arms. Lemma 2 also guarantees concentration of any statistical estimates for each arm as $t \to \infty$. We next show that the sampling of INDSELFSPARRING will concentrate around the optimal arm.

Algorithm 3 INDSELFSPARRING

input m the number of arms drawn at each iteration, η the learning rate

1: For each arm $i = 1, 2, \cdots, K$, set $S_i = 0$, $F_i = 0$.
2: **for** $t = 1, 2, \ldots$ **do**
3: **for** $j = 1, \ldots, m$ **do**
4: For each arm $i = 1, 2, \cdots, K$, sample θ_i from $Beta(S_i + 1, F_i + 1)$
5: Select $i_j(t) := \text{argmax}_i\, \theta_i(t)$
6: **end for**
7: Play m arms $\{i_j(t)\}_j$, observe pairwise feedback matrix $R = \{r_{jk} \in \{0, 1, \emptyset\}\}_{m \times m}$
8: **for** $j, k = 1, \ldots, m$ **do**
9: **if** $r_{jk} \neq \emptyset$ **then**
10: $S_j \leftarrow S_j + \eta \cdot r_{jk}$, $F_j \leftarrow F_j + \eta(1 - r_{jk})$
11: **end if**
12: **end for**
13: **end for**

Theorem 1. *Under Approximate Linearity,* INDSELFSPARRING *converges to the optimal arm b_1 as running time $t \to \infty$:* $\lim_{t \to \infty} \mathbb{P}(b_t = b_1) = 1$.

Theorem 1 implies that INDSELFSPARRING is asymptotically no-regret. As $t \to \infty$, the Beta distribution for each arm i is converging to $P(b_i \succ b_1)$, which implies converging to only choosing the optimal arm.

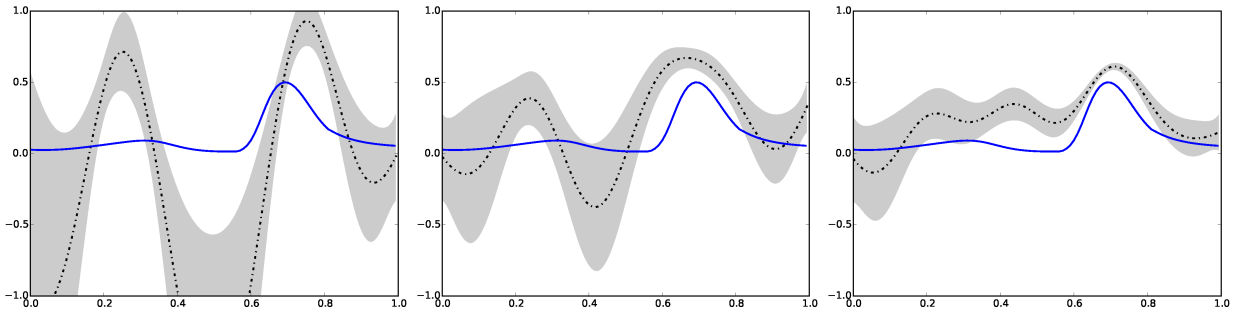
Most existing dueling bandits algorithm chooses one arm as a "reference" arm and the other arm as a competing arm for exploration/exploitation (in the $m = 2$ setting). If the distribution over reference arms never changes, then the competing arm is playing against a fixed "environment", i.e., it is a standard MAB problem. For general m, we can analogously consider choosing only one arm against a fixed distribution over all the other arms. Using Thompson sampling, the following lemma holds.

Lemma 3. *Under Approximate Linearity, selecting only one arm via Thompson sampling against a fixed distribution over the remaining arms leads to optimal regret w.r.t. choosing that arm.*

Lemma 3 and Theorem 1 motivate the idea of analyzing the regret of each individual arm against near-fixed (i.e., converging) environments.

Theorem 2. *Under Approximate Linearity,* INDSELFSPARRING *converges to the optimal arm with asymptotically optimal no-regret rate of $\mathcal{O}(K \ln(T)/\Delta)$.*

Theorem 2 shows an no-regret guarantee for INDSELF-



(a) 5 iterations (b) 20 iterations (c) 100 iterations

Figure 2: Evolution of a GP preference function in KERNELSELFSPARRING; dashed lines correspond to the mean and shaded areas to ± 2 standard deviations. The underlying utility function was sampled randomly from a GP with a squared exponential kernel with lengthscale parameter 0.2, and the resulting preference function is shown in blue. The GP finds the best arm with high confidence.

Algorithm 4 KERNELSELFSPARRING

input Input space S, GP prior (μ_0, σ_0), m the number of arms drawn at each iteration

for $t = 1, 2, \ldots$ **do**
  for $j = 1, \ldots, m$ **do**
    Sample f_j from $(\mu_{t-1}, \sigma_{t-1})$
    Select $i_j(t) := \operatorname{argmax}_x f_j(x)$
  end for
  Play m arms $\{i_j(t)\}_j$, observe pairwise feedback matrix $R = \{r_{jk} \in \{0, 1, \emptyset\}\}_{m \times m}$
  for $j, k = 1, \ldots, m$ **do**
    if $r_{jk} \neq \emptyset$ **then**
      apply Bayesian update using $(i_j(t), r_{jk})$ to obtain (μ_t, σ_t)
    end if
  end for
end for

SPARRING that asymptotically matches the optimal rate of $\mathcal{O}(K \ln(T)/\Delta)$ up to constant factors. In other words, once $t > C$ for some problem-dependent constant C, the regret of INDSELFSPARRING matches information-theoretic bounds up to constant factors (see Yue et al. (2012) for lower bound analysis).[3] The proof technique follows two major steps: (1) prove the convergence of INDSELFSPARRING as shown in Theorem 1; and (2) bound the expected total regret for sufficiently large T.

4.2 Dependent Arms Case

We use Gaussian processes (see Section 2.4) to model dependencies among arms. Applying Gaussian processes is not straightforward, since the underlying utility function is not directly observable or does not exist. We instead use Gaussian processes to model the preference function $f(b)$ corresponding to the preference of choosing b over the perfect "environment" of competing arms. Like in the independent arms case (Section 4.1), the perfect environment corresponds to having all the remaining arms be deterministically selected as the best arm b_1: $f(b) = P(b \succ b_1)$. We model $f(b)$ as a sample from a Gaussian process $GP(\mu(b), k(b, b'))$. Note that this setup is analogous to the independent arms case, which uses a Beta prior to estimate the probability of each arm defeating the environment (and converges to competing against the best environment).

Algorithm 4 describes KERNELSELFSPARRING, which instantiates SELFSPARRING using a Gaussian process Thompson sampling algorithm. The input space S can be continuous. At each iteration t, m arms are sampled using the Gaussian process prior D_{t-1}. The posterior D_t is then updated by the responses R and the prior.

Figure 2 illustrates the optimization process in a one-dimensional example. The underlying preference function against the best environment is shown in blue. Dashed lines are the mean function of GP. Shaded areas are ± 2 standard deviations regions (high confidence regions). Figures 2(a)(b)(c) represent running KERNELSELFSPARRING algorithm at 5, 20, and 100 iterations. The GP model can be observed to be converging to the preference function against the best environment.

We conjecture that it is possible to prove no-regret guarantees that scale w.r.t. the dimensionality of the kernel. However, there does not yet exist suitable regret analyses for Gaussian Process Thompson Sampling in the kernelized MAB setting to leverage.

[3] A finite-time guarantee requires more a refined analysis of C, and is an interesting direction for future work.

Name	Distribution of Utilities of arms
1good	1 arm with utility 0.8, 15 arms with utility 0.2
arith	1 arm with utility 0.8, 15 arms forming an arithmetic sequence between 0.7 and 0.2

Table 1: 16-arm synthetic datasets used for experiments.

5 EXPERIMENTS

5.1 Simulation Settings & Datasets

Synthetic Functions. We evaluated on a range of 16-arm synthetic settings derived from the utility-based dueling bandits setting of Ailon et al. (2014). For the multi-dueling setting, we used the following preference functions:

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

and the utility functions shown in Table 1 (generalized from those in Ailon et al. (2014)). Note that although these preference functions do not satisfy approximate linearity over their entire domains, they do for the utility samples (over the a finite subset of arms).

MSLR Dataset. Following the evaluation setup of Brost et al. (2016), we also used the Microsoft Learning to Rank (MSLR) WEB30k dataset, which consists of over 3 million query-document pairs labeled with relevance scores (Liu et al., 2007). Each pair is scored along 136 features, which can be treated as rankers (arms). For any subset of arms, we can estimate a preference matrix using the expected probability over the entire dataset of one arm beating another using top-10 interleaving and a perfect-click model. We simulate user feedback by using team-draft multileaving (Schuth et al., 2014).

5.2 Vanilla Dueling Bandits Experiments

We first compare against the vanilla dueling bandits setting of dueling a single pair of arms at a time. These experiments are included as a sanity check to confirm that SELFSPARRING (with $m = 2$) is a competitive algorithm in the original dueling bandits setting, and are not the main focus of our empirical analysis.

We empirically evaluate against a range of conventional dueling bandit algorithms, including:

- **Interleaved Filter (IF)** (Yue et al., 2012)
- **Beat the Mean (BTM)** (Yue & Joachims, 2011)
- **RUCB** (Zoghi et al., 2014)
- **MergeRUCB** (Zoghi et al., 2015b)
- **Sparring + UCB1** (Ailon et al., 2014)
- **Sparring + EXP3** (Dudík et al., 2015)
- **RMED1** (Komiyama et al., 2015)
- **Double Thompson Sampling** (Wu & Liu, 2016)

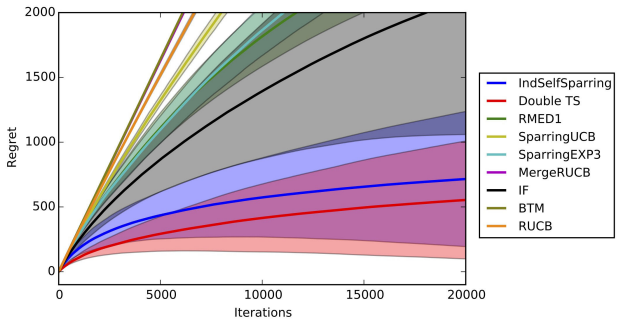


Figure 3: Vanilla dueling bandits setting. Average regret for top nine algorithms on logit/arith. Shaded regions correspond to one standard deviation.

For Double Thompson Sampling and INDSELFSPARRING, we set the learning rates to be 2.5 and 3.5 as optimized over a separate dataset of uniformly sampled utility functions. We use $\alpha = 0.51$ for RUCB/MergeRUCB, $\gamma = 1$ for BTM, and $f(K) = 0.3K^{1.01}$ for RMED1.

Results. For each scenario, we run each algorithm 100 times for 20000 iterations. For brevity, we show in Figure 3 the average regret of one synthetic simulation along with shaded one standard-deviation areas. We observe that SELFSPARRING is competitive with the best performing methods in the original dueling bandits setting. More complete experiments that replicate Ailon et al. (2014) are provided in the supplementary material, and demonstrate the consistency of this result.

Double Thompson Sampling (DTS) is the best performing approach in Figure 3, which is a fairly consistent result in the extended results in the supplementary material. However, given their high variances they are essentially comparable w.r.t. all other algorithms. Furthermore, INDSELFSPARRING has the advantage of being easily extensible to the more realistic multi-dueling and kernelized settings, which is not true of DTS.

5.3 Multi-Dueling Bandits Experiments

We next evaluate the multi-dueling setting with independent arms. We compare against the main existing approaches that are applicable to the multi-dueling setting, including the MDB algorithm (Brost et al., 2016), and the multi-dueling extension of Sparring, which we refer to as MultiSparring (Ailon et al., 2014). Following

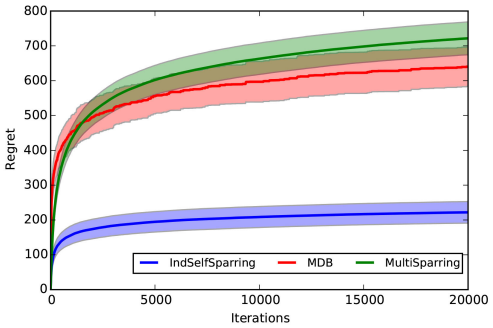


Figure 4: Multi-dueling regret for linear/1good setting

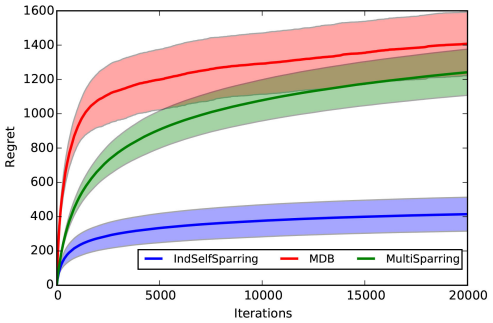


Figure 5: Multi-dueling regret for linear/arith setting

Brost et al. (2016), we use $\alpha = 0.5$ and $\beta = 1.5$ for the MDB algorithm. For INDSELFSPARRING, we set learning rate to be the default 1. Note that the vast majority dueling bandits algorithms are not easily applicable to the multi-dueling setting. For instance, RUCB-style algorithms treat the two arms asymmetrically, which is not easily generalized to multi-dueling.

Results on Synthetic Experiments. We test $m = 4$ on the linear 1good and arith datasets in Figure 4 and Figure 5, respectively. We observe that INDSELFSPARRING significantly outperforms competing approaches.

Results on MSLR Dataset. Following the simulation setting of Brost et al. (2016) on the MSLR dataset (see Section 5.1), we compared against the MDB algorithm over the same collection of 50 randomly sampled 16-arm subsets. We ensured that each 16-arm subset had a Condorcet winner; in general it is likely for any random subset of arms in the MSLR dataset to have a Condorcet winner (Zoghi et al., 2015a). Figure 6 shows the results, where we again see that INDSELFSPARRING enjoys significantly better performance.

5.4 Kernelized (Multi-)Dueling Experiments

We finally evaluate the kernelized setting for both the 2-dueling and the multi-dueling case. We evaluate KERNELSELFSPARRING against BOPPER (Gonzalez et al., 2016) and Sparring (Ailon et al., 2014) with GP-UCB

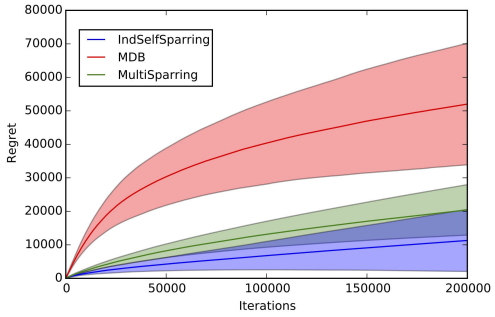


Figure 6: Multi-dueling regret for MSLR-30K experiments

(Srinivas et al., 2010). BOPPER is a Bayesian optimization method can be applied to kernelized 2-dueling setting (but not multi-dueling). Sparring with GP-UCB, which refer to as GP-Sparring, is essentially a variant of our KERNELSELFSPARRING approach but maintains a m GP-UCB bandit algorithms (one controlling each choice of arm to be dueled), rather than just a single one.

KERNELSELFSPARRING and GP-Sparring use GPs that model the preference function, i.e. are one-sided, whereas BOPPER uses a GP to model the entire preference matrix. Following Srinivas et al. (2010), we use a squared exponential kernel with lengthscale parameter 0.2 for both GP-Sparring and KERNELSELFSPARRING, and use a squared exponential kernel with parameter 1 for BOPPER. We initialize all GPs with a zero-mean prior, and use sampling noise variance $\sigma^2 = 0.025$. For GP-Sparring, we use the scaled-down version of β_t as suggested by Srinivas et al. (2010).

We use the Forrester and Six-Hump Camel functions as utility functions on $[0, 1]$ and $[0, 1]^2$, respectively, as in Gonzalez et al. (2016). Similarly, we use the same uniform discretizations of 30 and 64 points for the Forrester and Six-Hump Camel settings respectively, and use the logit link function to generate preferences.

Since the BOPPER algorithm is computationally expen-

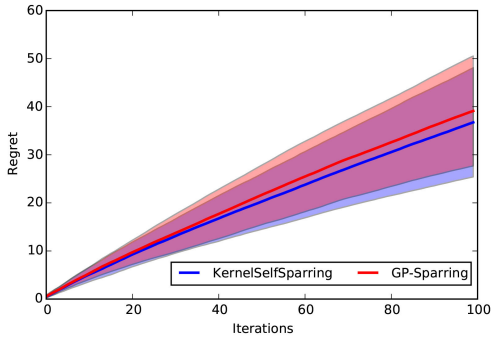


Figure 7: 2-dueling regret for kernelized setting with synthetic preferences

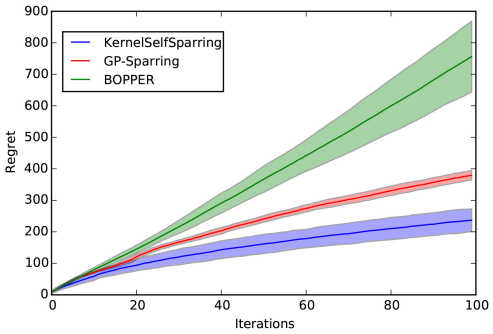


Figure 8: 2-dueling regret for kernelized setting with Forrester objective function

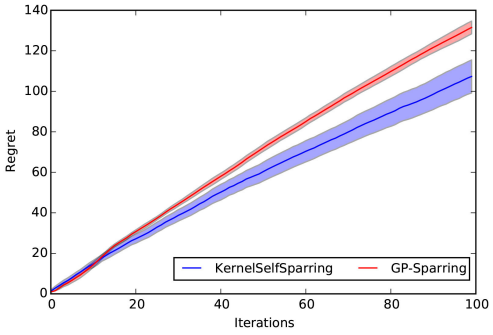


Figure 9: 2-dueling regret for kernelized setting with Six-Hump Camel objective function

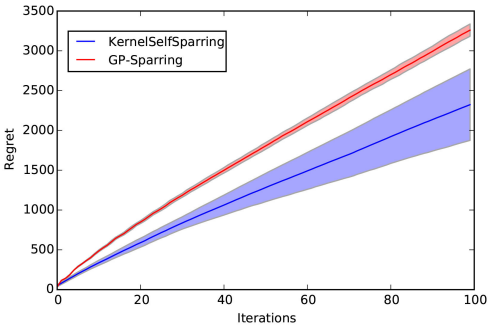


Figure 10: Multi-dueling regret for kernelized setting with Forrester objective function

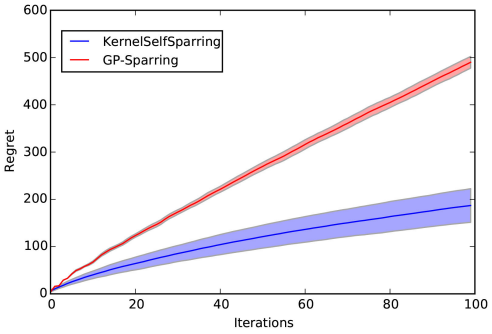


Figure 11: Multi-dueling regret for kernelized setting with Six-Hump Camel objective function

sive, we only include it in the Forrester setting, and run each algorithm 20 times for 100 iterations. In the Six-Hump Camel setting, we run KERNELSELFSPARRING and GP-Sparring for 500 iterations 100 times each. Results are presented in Figures 8 and 9, where we observe much better performance from KERNELSELFSPARRING against both BOPPER and GP-Sparring.

In the kernelized multi-dueling setting, we compare against GP-Sparring. We run each algorithm for 100 iterations 50 times on the Forrester and Six-Hump Camel functions, and plot their regrets in Figures 10 and 11 respectively. We use $m = 4$ for both algorithms, and the same discretization as in the standard dueling case. We again observe significant performance gains of our KERNELSELFSPARRING approach.

6 CONCLUSIONS

We studied multi-dueling bandits with dependent arms. This setting extends the original dueling bandits setting by dueling multiple arms per iteration rather than just two, and modeling low-dimensional dependencies between arms rather than treat each arm independently. Both extensions are motivated by practical real-world considerations such as in personalized clinical treatment (Sui & Burdick, 2014). We proposed SELFSPARRING, which is simple and easy to extend, e.g., by integrating with kernels to model dependencies across arms. Our experimental results demonstrated significant reduction in regret compared to state-of-the-art dueling bandit algorithms. Generally, relative benefits compared to dueling bandits increased with the number of arms being compared. For SELFSPARRING, the incurred regret did not increase substantially as the number of arms increased.

Our approach can be extended in several important directions. Most notably, the theoretical analysis could be improved. For instance, it would be more desirable to provide explicit finite-time regret guarantees rather than asymptotic ones. Furthermore, an analysis of the kernelized multi-dueling setting is also lacking. From a more practical perspective, we assumed that the choice of arms does not impact the feedback mechanism (e.g., all pairs), which is not true in practice (e.g., humans can have a hard time distinguishing very different arms).

Acknowledgments. This work was funded in part by NSF Awards #1564330 & #1637598, JPL PDF IAMS100224, a Bloomberg Data Science Research Grant, and a gift from Northrop Grumman.

References

Agrawal, Shipra and Goyal, Navin. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory (COLT)*, 2012.

Ailon, Nir, Karnin, Zohar, and Joachims, Thorsten. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning (ICML)*, 2014.

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

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

Brost, Brian, Seldin, Yevgeny, Cox, Ingemar J, and Lioma, Christina. Multi-dueling bandits and their application to online ranker evaluation. In *ACM Conference on Information and Knowledge Management*, 2016.

Chapelle, Olivier and Li, Lihong. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Chapelle, Olivier, Joachims, Thorsten, Radlinski, Filip, 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.

Dudík, Miroslav, Hofmann, Katja, Schapire, Robert E, Slivkins, Aleksandrs, and Zoghi, Masrour. Contextual dueling bandits. In *Conference on Learning Theory (COLT)*, 2015.

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

Gonzalez, Javier, Dai, Zhenwen, Damianou, Andreas, and Lawrence, Neil D. Bayesian optimisation with pairwise preferential returns. In *NIPS Workshop on Bayesian Optimization*, 2016.

Hofmann, Katja, Whiteson, Shimon, and De Rijke, Maarten. A probabilistic method for inferring preferences from clicks. In *ACM Conference on Information and Knowledge Management*, 2011.

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

Kaufmann, Emilie, Korda, Nathaniel, and Munos, Rémi. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory (ALT)*, 2012.

Komiyama, Junpei, Honda, Junya, Kashima, Hisashi, and Nakagawa, Hiroshi. Regret lower bound and optimal algorithm in dueling bandit problem. In *COLT*, pp. 1141–1154, 2015.

Liu, Tie-Yan, Xu, Jun, Qin, Tao, Xiong, Wenying, and Li, Hang. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR 2007 workshop on learning to rank for information retrieval*, pp. 3–10, 2007.

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

Russo, Daniel and Van Roy, Benjamin. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Schuth, Anne, Sietsma, Floor, Whiteson, Shimon, Lefortier, Damien, and de Rijke, Maarten. Multileaved comparisons for fast online evaluation. In *ACM Conference on Conference on Information and Knowledge Management*, 2014.

Schuth, Anne, Oosterhuis, Harrie, Whiteson, Shimon, and de Rijke, Maarten. Multileave gradient descent for fast online learning to rank. In *ACM Conference on Web Search and Data Mining*, 2016.

Srinivas, Niranjan, Krause, Andreas, Kakade, Sham, and Seeger, Matthias. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

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

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

Wu, Huasen and Liu, Xin. Double thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems*, 2016.

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

Yue, Yisong and Joachims, Thorsten. Beat the mean bandit. In *International Conference on Machine Learning (ICML)*, 2011.

Yue, Yisong, Broder, Josef, Kleinberg, Robert, and Joachims, Thorsten. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zoghi, Masrour, Whiteson, Shimon, Munos, Remi, and de Rijke, Maarten. Relative upper confidence bound for the k-armed dueling bandit problem. In *International Conference on Machine Learning (ICML)*, 2014.

Zoghi, Masrour, Karnin, Zohar S, Whiteson, Shimon, and de Rijke, Maarten. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pp. 307–315, 2015a.

Zoghi, Masrour, Whiteson, Shimon, and de Rijke, Maarten. Mergerucb: A method for large-scale online ranker evaluation. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2015b.

A Proofs

This section provides the proof sketch of Lemmas and Theorems mentioned in the main paper.

Lemma 1. For the K-armed stochastic MAB problem, Thompson Sampling has expected regret: $\mathbb{E}[R_T^{\text{MAB}}] = \mathcal{O}\left(\frac{K}{\Delta}\ln T\right)$, where Δ is the difference between expected rewards of the best two arms.

Proof. This lemma is a direct result from Theorem 2 of Agrawal & Goyal (2012) and Theorem 1 of Kaufmann et al. (2012). □

Lemma 2. Running INDSELFSPARRING with infinite time horizon will sample each arm infinitely often.

Proof. Proof by contradiction.
Let $B(x;\alpha,\beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1}dt$. Then the CDF of Beta distribution with parameters (α,β) is

$$F(x;\alpha,\beta) = \frac{B(x;\alpha,\beta)}{B(1;\alpha,\beta)}.$$

Suppose arm b can only be sampled in finite number of iterations. Then there exists finite upper bound T_b for $\alpha_b + \beta_b$. For any given $x \in (0,1)$, the probability of sampling values of arm b θ_b greater than x is

$$P(\theta_b > x) = 1 - F(x;\alpha_b,\beta_b)$$

$$\geq 1 - F(x;1,T_b-1) = (1-x)^{T_b-1} > 0$$

Then by running INDSELFSPARRING, the probability of choosing arm b after it has been chosen T_b times:

$$P(\theta_b \geq max_i\{\theta_{b_i}\}) \geq \prod_i P(\theta_b \geq \theta_{b_i})$$

is strictly non-zero. That violates any fixed upper bound T_b. □

Theorem 1. Under Approximate Linearity, INDSELFSPARRING converges to the optimal arm b_1 as running time $t \to \infty$: $\lim_{t\to\infty}\mathbb{P}(b_t = b_1) = 1$.

Proof. INDSELFSPARRING keeps one Beta distribution $Beta(\alpha_i(t),\beta_i(t))$ for each arm b_i at time step t. Let $\hat{\mu}_i(t) = \frac{\alpha_i(t)}{\alpha_i(t)+\beta_i(t)}$, $\hat{\sigma}_i^2(t) = \frac{\alpha_i(t)\beta_i(t)}{(\alpha_i(t)+\beta_i(t))^2(\alpha_i(t)+\beta_i(t)+1)}$ be the empirical mean and variance for arm b_i.
Obviously, $\hat{\sigma}_i^2(t) \to 0$ as $(\alpha_i(t)+\beta_i(t)) = (S_i(t)+F_i(t)) \to \infty$. By Lemma 2 we have $(S_i(t)+F_i(t)) \to \infty$ as $t \to \infty$. That shows every Beta distribution is concentrating to a Dirac function at $\hat{\mu}_i(t)$ when $t \to \infty$.

Define $\hat{\mu}(t) = [\hat{\mu}_1(t),\cdots,\hat{\mu}_K(t)]^T \in [0,1]^K$to be the vector of means of all arms. Then $\mu = \{\mu_i = P(b_i \succ b_1)\}_{i=1,\cdots,K}$ is a stable point for INDSELFSPARRING in the K dimensional mean space.

Suppose there exists another stable point $\nu \in [0,1]^K (\nu \neq \mu)$ for INDSELFSPARRING, consider the following two possibilities: (1) $\nu_1 = max_i\{\nu_i\}$ and (2) $\nu_1 < max_i\{\nu_i\} = \nu_j$.

Since the Beta distributions for each arm b_i is concentrating to Dirac functions at ν_i, $P(\theta_i > \theta_j) \in [\mathbb{I}(\nu_i > \nu_j) - \delta, \mathbb{I}(\nu_i > \nu_j) + \delta]$ for any fixed $\delta > 0$ with high probability.

If (1) holds, then ν_1 will converge to $\frac{1}{2} = \mu_1$ and ν_i will converge to $P(b_i \succ b_1) = \mu_i$. Thus $\nu = \mu$. Contradict to $\nu \neq \mu$.

If (2) holds, then ν_j will converge to $\frac{1}{2} = \mu_1$ and $\nu_1 \in [P(b_1 \succ b_j) - \delta, P(b_1 \succ b_j) + \delta]$ for any fixed $\delta > 0$ with high probability. Since $P(b_1 \succ b_j) \geq \frac{1}{2} + \Delta$, $\nu_1 \in [P(b_1 \succ b_j) - \delta, P(b_1 \succ b_j) + \delta] \geq \frac{1}{2} + \Delta - \delta$. Since δ can be arbitrarily small, we have $\nu_1 \geq \frac{1}{2} + \Delta - \delta > \frac{1}{2} + \delta > \nu_j$. That contradict to $\nu_1 < \nu_j$.

In summary, $\mu = \{\mu_i = P(b_i \succ b_1)\}_{i=1,\cdots,K}$ is the only stable point in the mean space. As $\hat{\mu}(t) \to \mu$, $\mathbb{P}(b_t = b_1) \to 1$.

Define $\mathbb{P}_t = [P_1(t), P_2(t), ..., P_K(t)]$ as the probabilities of picking each arm at time t. Let $\mathbb{P} = \{\mathbb{P}_t\}_{t=1,2,...}$ be the sequence of probabilities w.r.t. time. Assume INDSELFSPARRING is non-convergent. It is equivalent to say that $\mathbb{P}$ is not converging to a fixed distribution. Then $\exists\delta > 0$ and arm i s.t. the sequence of probabilities $\{P_i(t)\}_t$ satisfies:

$$\limsup_{t\to\infty} P_i(t) - \liminf_{t\to\infty} P_i(t) > \delta$$

w.h.p. which is equivalent of having:

$$\limsup_{t\to\infty} \hat{\mu}_i(t) - \liminf_{t\to\infty} \hat{\mu}_i(t) > \epsilon$$

w.h.p. for some fixed $\epsilon > 0$. This violates the stability of INDSELFSPARRING in the K dimensional mean space as shown above. So as $t \to \infty$, $\hat{\mu}(t) \to \mu$, $\mathbb{P}(b_t = b_1) \to 1$.

□

Lemma 3. Under Approximate Linearity, selecting only one arm via Thompson sampling against a fixed distribution over the remaining arms leads to optimal regret w.r.t. choosing that arm.

Proof. We first prove the results for $m = 2$. Results for any $m > 2$ can be proved in a similar way.

Consider Player 1 drawing arms from a fixed distribution L. Player 2's drawing strategy is an MAB algorithm $\mathcal{A}$.

Let $R_A(T)$ be the regret of algorithm $\mathcal{A}$ within horizon T. $B(T) = \sup \mathbb{E}[R_A(T)]$ is the supremum of the expected regret of $\mathcal{A}$.

The reward of Player 2 at iteration t is $\phi(b_{2t}, b_{1t})$. Reward of keep playing the optimal arm is $\phi(b_1, b_{1t})$. So the total regret after T rounds is

$$R_A(T) = \sum_{t=1}^{T} [\phi(b_1, b_{1t}) - \phi(b_{2t}, b_{1t})]$$

Since Approximate Linearity yields

$$\phi(b_1, b_{1t}) - \phi(b_{2t}, b_{1t}) \geq \gamma \cdot \phi(b_1, b_{2t})$$

We have

$$\mathbb{E}[R_A(T)] = \mathbb{E}\mathbb{E}_{b_{1t} \sim L} \left[\sum_{t=1}^{T} [\phi(b_1, b_{1t}) - \phi(b_{2t}, b_{1t}) \right]$$

$$\geq \mathbb{E}\mathbb{E}_{b_{1t} \sim L} \left[\sum_{t=1}^{T} \gamma \cdot \phi(b_1, b_{2t}) \right]$$

$$= \gamma \cdot \mathbb{E} \left[\sum_{t=1}^{T} \phi(b_1, b_{2t}) \right] = \gamma \cdot \mathbb{E}[R(T)]$$

So the total regret of Player 2 is bounded by

$$\mathbb{E}[R(T)] \leq \frac{1}{\gamma} \mathbb{E}[R_A(T)] \leq \frac{1}{\gamma} \sup \mathbb{E}[R_A(T)] = \frac{1}{\gamma} B(T)$$

□

Corollary 1. *If approximate linearity holds, competing with a drifting but converging distribution of arms guarantees the one-side convergence for Thompson Sampling.*

Proof. Let D_t be the drifting but converging distribution and $D_t \to D$ as $t \to \infty$. Let b_T be the drifting mean bandit of D_T after T iterations. Since D_t is convergent, $\exists T > K$ such that

$$\phi(\sup_{t>T} b_T, \inf_{t>T} b_T) < \phi(b_1, b_2)$$

where $\phi(b_1, b_2)$ is the preference between the best two arms. The mean value of feedback by playing arm i is $\phi(b_i, b_T)$. If b_T is fixed, by Lemma3, Thompson sampling converges to the arm: $i^* = \text{argmax}_i \, \phi(b_i, b_T)$. For drifting b_T, define $b^+ = \sup_{t>T} b_T$ and $b^- = \inf_{t>T} b_T$.

Thompson sampling convergence to the optimal arm implies that:

$$\phi(b_1, b^+) > \phi(b_i, b^-)$$

for all $i \neq 1$. Consider:

$$\begin{aligned} &\phi(b_1, b^+) - \phi(b_2, b^-) \\ &= \phi(b_1, b^+) - \phi(b_2, b^-) + \phi(b_1, b^-) - \phi(b_1, b^-) \\ &= \phi(b_1, b^-) - \phi(b_2, b^-) + \phi(b_2, b^+) - \phi(b_1, b^-) \\ &\geq \gamma \cdot [\phi(b_1, b_2) - \phi(b^+, b^-)] > 0 \end{aligned}$$

by approximate linearity.

So we have $\phi(b_1, b^+) > \phi(b_2, b^-)$. Since $\phi(b_2, b^-) > \phi(b_i, b^-)$ for $i > 2$. Then we have

$$\phi(b_1, b^+) > \phi(b_i, b^-)$$

holds for all $i \neq 1$. So Thompson sampling converge to the optimal arm. □

Theorem 2. Under Approximate Linearity, IndSelfSparring converges to the optimal arm with asymptotically optimal no-regret rate of $\mathcal{O}(K \ln(T)/\Delta)$. Where Δ is the difference between the rewards of the best two arms.

Proof. Theorem 1 provides the convergence guarantee of IndSelfSparring. Corollary 1 shows one-side convergence for playing against a converging distribution.

Since IndSelfSparring converges to the optimal arm b_1 as running time $t \to \infty$: $\lim_{t\to\infty} \mathbb{P}(b_t = b_1) = 1$. For $\forall \delta > 0$, there exists $C(\delta) > 0$ such that for any $t > C(\delta)$, the following condition holds w.h.p.: $P(b_t = b_1) \geq 1 - \delta$.

For the triple of bandits $b_1 \succ b_i \succ b_K$, Approximate Linearity guarantees:

$$\phi(b_i, b_K) < \phi(b_1, b_K) \leq \omega$$

holds for some fixed $\omega > 0$ and $\forall i \in \{2, \cdots, K-1\}$. With small δ, the competing environment of any Player p is bounded. If $\delta < \frac{\Delta}{\Delta+\omega}$, $(1-\delta)\cdot(-\Delta)+\delta\cdot\phi(b_2, b_K) < 0 = 1 \cdot \phi(b_1, b_1)$. The competing environment can be considered as unbiased and the theoretical guarantees for Thompson sampling for stochastic multi-armed bandit is valid (up to a constant factor).

Then IndSelfSparring has an no-regret guarantee that asymptotically matches the optimal rate of $\mathcal{O}(K \ln(T)/\Delta)$ up to constant factors, which proves Theorem 2.

□

B Further Experiments

We run further experiments on 16-arm synthetic datasets. The distributions of utilities of arms are shown in Table 2. We compared the performances of 8 algorithms and 15 scenarios as shown in Figure 12.

Name	Distribution of Utilities of arms
1good	1 arm with utility 0.8, 15 arms with utility 0.2
2good	1 arm with utility 0.8, 1 arms with utility 0.7, 14 arms with utility 0.2
6good	1 arm with utility 0.8, 5 arms with utility 0.7, 10 arms with utility 0.2
arith	1 arm with utility 0.8, 15 arms forming an arithmetic sequence between 0.7 and 0.2
geom	1 arm with utility 0.8, 15 arms forming a geometric sequence between 0.7 and 0.2

Table 2: 16-arm synthetic datasets used for experiments.

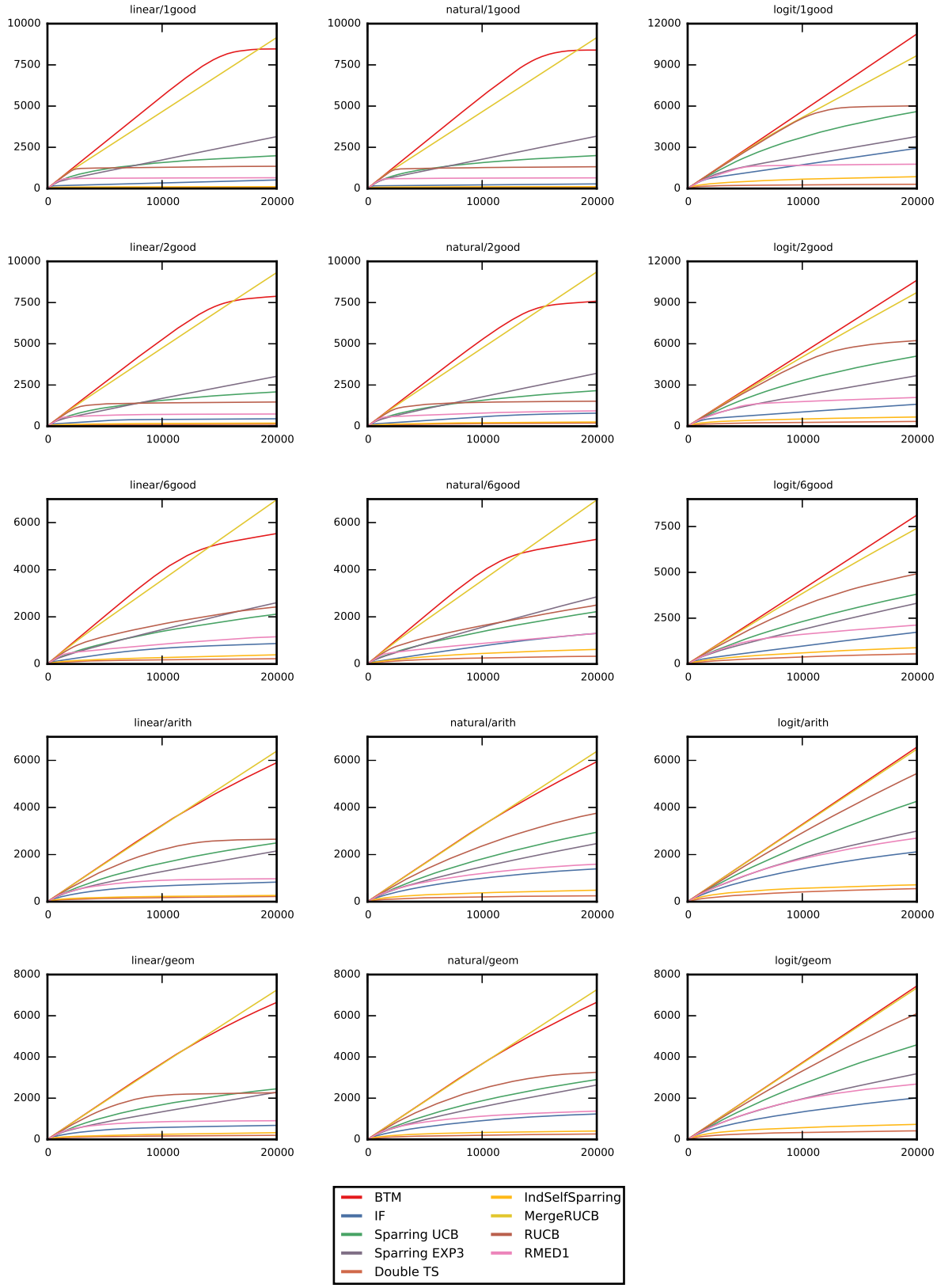


Figure 12: Average regret vs iterations for each of 8 algorithms and 15 scenarios.