Battle of Bandits

Aadirupa Saha
aadirupa@iisc.ac.in
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore 560012

Aditya Gopalan
aditya@iisc.ac.in
Dept. of Electrical Communication Engineering
Indian Institute of Science, Bangalore 560012

Abstract

We introduce *Battling-Bandits* – an online learning framework where given a set of n arms, the learner needs to select a subset of $k \geq 2$ arms in each round and subsequently observes a stochastic feedback indicating the winner of the round. This framework generalizes the standard *Dueling-Bandit* framework which applies to several practical scenarios such as medical treatment preferences, recommender systems, search engine optimization etc., where it is easier and more effective to collect feedback for multiple options simultaneously. We develop a novel class of *pairwise-subset choice model*, for modelling the subset-wise winner feedback and propose three algorithms - *Battling-Doubler*, *Battling-MultiSBM* and *Battling-Duel*: While the first two are designed for a special class of *linear-link* based choice models, the third one applies to a much general class of *pairwise-subset choice models* with *Condorcet winner*. We also analyzed their regret guarantees and show the optimality of *Battling-Duel* proving a matching regret lower bound of $\Omega(n \log T)$, which (perhaps surprisingly) shows that the flexibility of playing size-k subsets does not really help to gather information faster than the corresponding dueling case ($k = 2$), at least for the current subsetwise feedback choice model. The efficacy of our algorithms are demonstrated through extensive experimental evaluations on a variety of synthetic and real world datasets.

1 INTRODUCTION

The problem of *Dueling-Bandits* has recently gained much attention in the machine learning community [22, 4, 24, 25, 23, 16]. Dueling bandits is an online learning framework, generalizing multi-armed bandits [5], in which learning proceeds in rounds: At each round the learner selects a pair of arm and observes a stochastic feedback of the winner of the comparison (duel) between the selected arms. Several algorithms have been proposed for this problem which are designed to learn to play the best arm as often as possible over time [22, 4, 24, 25, 23, 16]. These algorithms are tailor-made to work well under specific assumptions on the underlying pairwise comparison model that generates the stochastic feedback and under specific definition of the winner of a set of arms [16].

In this work, we introduce a natural generalization of the dueling bandits problem where given a set of n items (bandit arms), the learner's objective at each round is to choose a subset of $k \geq 2$ arms (unlike selecting just *two* arms as in case of dueling bandits), upon which the winner of the *'battle'* among these k selected items is revealed by the environment as stochastic feedback. The goal of the learner is to identify an appropriately defined *'best'* item in the process and play it often as possible.

We term this as the problem of *Battling-Bandits* as at each round, essentially a subset of k items are competing against each other unlike a pairwise duel as in the case of *Dueling-Bandits*. Such settings occur naturally in many application domains where it is practically easier for customers or patients to give a *single* feedback for a set of options (products or medical treatments), click on one link from set of search engine outcomes etc., as opposed to comparing only two options at a time. To the best of our knowledge this is the first work to generalize the pairwise feedback model of dueling bandits to a subsetwise model in an online regret minimization setup.

Related Work

The most related work to the current problem setup is [17], where also a fixed set of arms in chosen in each round. However, the key difference lies in the feedback structure. While in [17] the feedback is a pairwise preference matrix consisting outcomes of one or more chosen pairs (maximum of $\binom{n}{2}$ pairs), in our setting we only observe a single index of the winning arm. In [8], the authors also consider an extension of the dueling bandits framework where multiple arms are chosen in each round. We differ from their setup since we allow to choose only a fixed k-set of arms at each round, whereas [8] allows a variable number of arm selection. Moreover their work does not have any theoretical guarantees while we provide regret guarantees for our algorithms. Another work in the similar essence is DCM-bandits [12], where a list of k distinct items are offered at each round and the users choose one or more from it scanning the list from top to bottom. Their learning objective differs substantially from ours since the DCM feedback is based on a fairly different cascading feedback model. Moreover, their regret objective demands to find the set of best k items as opposed to finding a unique best item as in our case.

Another related body of literature is dynamic assortment optimization where the objective is offer a subset of a fixed set of items to the customers in order to maximize the expected revenue. The demand of any item depends on the substitution behavior of the customers that is captured mathematically by a choice model specifying the probability of a consumer selecting a particular item from any offered set. The problem has been studied under different choice models – e.g. multinomial logit [19], mallows and mixture of mallows [9], markov chain based choice models [10], single transition model [15], general discrete choice models [7] etc. A related bandit setting has also been studied as the MNL-Bandits problem in [2] where the learner selects a fixed set of k arms in each iteration. However, the feedback is observed from a multinomial logit model (MNL) which is different from the subset choice model we considered here. Moreover their setting takes item prices into account due to which the notion of the *'best item'* is different from ours, i.e. the *Condorcet winner*. Thus our current problem setting can not be reduced to theirs and vice-versa.

Proposed Work

The main challenge of *Battling-Bandits* lies in keeping track of the subset choice probabilities, i.e. the probability of an item winning in a given subset of k items, which could be potentially of size $O(kn^k)$ as our objective is to find the "best" (Condorcet winner) item in the hindsight, we must allow repetitions of items within a offered set, which actually results in n^k possible number of subsets and each subset may give rise to atmost k choice probabilities depending on number of distinct items in the subset. Thus without any further structural or parametric assumptions on the feedback choice model, the problem becomes computationally intractable.

We thus introduce the *pairwise-subset choice model* for the purpose which is based on a pairwise preference model with Condorcet winner (Section 2) and propose three different algorithms (Section 3): The first two – *Battling-Doubler* and *Battling-MultiSBM*, are inspired by the Doubler and MultiSBM algorithms of [4] which works under a special class of *pairwise-subset choice model*, viz. *linear-subset choice model*, which naturally generalizes the linear-link based dueling feedback model of [4]. Both the algorithms are based on a novel reduction of the *battling bandit* problem to classical *multiarmed bandit* (MAB) [5]. Note that, although they apply to a special subclass of choice models, their regret guarantees hold for a richer class of arm sets, e.g. the regret of *Battling-Doubler* holds for any general class of (even infinitely many!) structured arms, whereas *Battling-MultiSBM* applies to any finite set of unstructured arms.

Our third algorithm, *Battling-duel*, works for the most general class of *pairwise subset choice models*, which is built on the novel idea of reducing *battling bandits* to the *dueling* case by using a dueling bandit algorithm as a black box, e.g. Relative-UCB [24] or Double-Thompson Sampling [21] which are guaranteed to work optimally (with $O(n \log T)$ regret guarantee) under any pairwise preference based feedback model with Condorcet winner.

Contributions. The specific contributions of this paper can be summarized as follows:

1. We develop a novel class of subsetwise feedback model, called *pairwise-subset choice model*, which is based on a pairwise preference model with Condorcet winner that models the winning probability of an item in a battle in terms of its pairwise winning probabilities over others. We further analyse a special class of the above model, namely *linear-subset choice model* which generalizes the linear-link based dueling feedback model of [4] (Section 2).

2. We propose three algorithms for the probelme of *Battling-Bandits* and analyze the regret guarantees of each under a natural notion of regret with respect to the Condorcet item (see Section 2.2). In particular, we show that the regret for the first two algorithms, *Battling-Doubler* and *Battling-MultiSBM*, scales as $O(nk\ln^2(T))$ and $O(nK(\ln(T) + n\ln(n) + n\ln\ln(T))$ respectively, under *linear-subset choice model*. The regret of our third algorithm *Battling-Duel* holds under the general class of *pairwise-subset choice model* that scales as $O(n \ln T)$ (Section 3).

3. We also prove a lower bound of $\Omega(n\ln(T))$ for *Battling-Bandits* under *pairwise-subset choice model* which shows that the regret of *Battling-Duel* algorithm matches the lower bound (upto constant factor),

thereby making it the optimal possible algorithm for the current problem setup (Section 4). An interesting and perhaps surprising point to note here is that our regret bounds are independent of the subset size k, which implies the flexibility of playing larger subsets does not really help to gather information faster than the corresponding dueling case ($k = 2$), atleast with the current setting of the battling problem.

4. Our extensive simulation based experiments justifies the derived theoretical guarantees of our proposed algorithms. We also compare our algorithms to *Self-Sparring* algorithm of [17], which is the only existing work applicable to our setting and show the superior performance of our algorithms on both synthetic and real word data sets (Section 5).

Organization: In Section 2, we describe the problem setup and introduce our notions of regret. Section 3 describes our three proposed algorithms along with theoretical regret guarantees. In Section 4, we derive the lower bound for *Battling-Bandits* problem. Section 5 presents our experimental evaluations and finally we conclude with remarks and directions for future work in Section 6.

2 PROBLEM SETUP

We proposed the problem of *Battling-Bandit* (or in short BB) as a natural generalization of the well-studied *Dueling-Bandit (DB)* problem in the bandit literature: Given a set of $n \geq 2$ items (equivalently, bandit arms) denoted by $[n] = \{1, 2, \ldots, n\}$, at each round $t \in \mathbb{N}$, the learner's task is to build a multiset of $k \geq 2$ items from $[n]$. The environment then picks a 'winner' – one of the k items from the chosen set – according to a subset choice model, unknown to the learner, and reveals the winner's identity to the learner. We denote by $S_t \subseteq [n]$ the multiset of k items chosen by the learner, i.e., $S_t \equiv (S_t(1), \ldots, S_t(k)) \in [n]^k$, and $i_t^* \in [k]$ to be the index of the winning item in S_t, at iteration t. Each selection of k items also carries with it a cost or regret. The aim of the learner is to choose sets of items to minimize the total cumulative regret over a time horizon T.

From a different point of view, the setting of receiving the winner information of the subset S_t at each round t can be seen as a game between k players. Each player is associated with an index $i \in [k]$ and chooses an arm from $[n]$, thus specifying the multiset S_t. The winning player is the index of the winning item revealed to the learner at time t – the winner of the battle among k players. Hence we named it as the problem of *Battling-Bandit (BB)*. We next describe the rule of winner selection in a given battle.

2.1 Subset Choice Models

Given a fixed set of items (context), choice modeling defines the decision probability of preferring an individual or set of items through stochastic models. In the present case, we use subset choice models to specify the winning probability of an item in a given set. We first introduce a broad class of subset choice models, called *pairwise-subset choice models*, extending the notions from pairwise preference models for the dueling bandit ($k = 2$) problem.

Pairwise-subset choice model. We define a class of subset choice models based on any pairwise preference matrix $\mathbf{Q} \in [0, 1]^{n \times n}$, where $Q_{a,b}$ denotes the probability of arm a beating b, for any $a, b \in [n]$. Clearly, $Q_{a,b} + Q_{b,a} = 1$. Now given a set $S \subseteq [n]$ of k items with $S \equiv (a_1, \ldots, a_k) \in [n]^k$ and any $i \in [k]$, we define the probability of i^{th} index gets selected as the winner as:

$$P(i|S) = \sum_{j=1, j\neq i}^{k} \frac{2Q_{a_i,a_j}}{k(k-1)} \ \ \forall i \in [k]. \tag{1}$$

It can be easily checked that the formula above defines a valid probability distribution over the indices $i \in [k]$. We remark that since S is a multiset, the arm corresponding to the winning index is not necessarily unique; as an extreme example, in the multiset of k items $(a_1, a_2, \ldots, a_k)$, we might have $a_i = a \in [n]$, $\forall i \in [k]$, in which case each index $i \in [k]$ wins with probability $1/k$.

Note that when $k = 2$ (the dueling bandit case), for any $S = (a, b)$, we have $P(i|S) = Q_{a_i,a_j}$, where $i, j \in [2], i \neq j$ and $a_1 = a$ and $a_2 = b$; which defines the pairwise probability of item a winning over item b in a pairwise duel. The following result provides an alternative interpretation of the *pairwise-subset choice model* in terms of the average probability that the item in question wins in a randomly chosen duel:

Lemma 1. *Let $S \equiv (a_1, \ldots, a_k) \in [n]^k$ be a multiset of k arms from $[n]$. Suppose U and V are two items (indices) chosen uniformly at random without replacement from $[k]$, and $W \in [2]$ is drawn as the winning index according to the pairwise preference model $\mathbf{Q}$ over the set (a_U, a_V). Let $X = U$ if $W = 1$ and $X = V$ if $W = 2$. Then, for each $i \in [k]$, $\mathbf{P}(i|S)$ in (1) is the probability that $X = i$.*

Remark 1. Note that if a *Condorcet winner* [16] $a^* \in [n]$ exists with respect to the preference matrix $\mathbf{Q}$, i.e. $\exists a^* \in [n]$, such that $Q_{a^*,j} > \frac{1}{2}$, $\forall j \in [n] \setminus \{a^*\}$, then it is easy to verify that for any (multi)set $S \subseteq [n]$, $P(i|S) > P(j|S)$ whenever $a_i = a^*$ and $a_j \in [n] \setminus \{a^*\}$, $\forall i, j \in [k], i \neq j$. Our objective is to identify this 'best' arm a^* and play it as often as possible; as spelt out in the definition of our regret (Section 2.2).

We now define an utility score based subset choice model as a special class of *pairwise-subset choice models*.

Linear-subset choice model. Let us assume that each arm $a \in [n]$ is associated with an unknown utility score $\theta_a \in [0,1]$. Then given a multiset of k items $S \equiv (a_1, \ldots, a_k) \in [n]^k$, the probability that its i^{th} index gets selected as the winner with probability

$$\mathbf{P}(i|S) = \frac{\sum_{j=1, j\neq i}^{k}(\theta_{a_i} - \theta_{a_j} + 1)}{k(k-1)}$$
$$= \frac{1}{k} + \frac{\sum_{j=1, j\neq i}^{k}(\theta_{a_i} - \theta_{a_j})}{k(k-1)}, \forall i \in [k]. \quad (2)$$

We call this the *linear-subset choice model* since the model is can be seen as a special case of *pairwise-subset choice model* when the underlying pairwise preference model $\mathbf{Q}^{\boldsymbol{\theta}}$ is linear, i.e. $\Pr(a \text{ beats } b) = Q^{\boldsymbol{\theta}}_{a,b} = \frac{(\theta_a - \theta_b + 1)}{2}$, $a, b \in [n]$. Note that this model generalizes the linear-link based pairwise feedback model of [4] at $k = 2$, as for any $S = (a,b)$, the probability $\Pr(a \text{ beats } b) = (\theta_a - \theta_b + 1)/2$ becomes exactly equal to that of [4] used for modeling the dueling bandit feedback.

Remark 2. The *linear-subset choice model* satisfy a natural monotonicity property: For any set $S \equiv (a_1, \ldots, a_k)$ and $i, j \in [k]$, $\theta_{a_i} > \theta_{a_j} \Rightarrow P(i|S) > P(j|S)$, thus the element with highest θ score is most likely to get selected as the winner of set S. In other words, an ordering over θ values, induces an ordering over the arms as well.

There also exist other notions subset choice models in the literature, e.g., one popular class among them is the *random utility based* models (RUM) [6] as described below:

RUM based Choice Models. One of the most popularly studied class of choice models are *Random Utility Models (RUM)*. RUM assumes an underlying ground truth of utility score $\theta_i \in \mathbb{R}$ for each item $i \in [n]$, and assigns a conditional distribution $\mathcal{D}_i(\cdot|\theta_i)$ for scoring item i. So given $S \subseteq [n]$, one first draws a random utility score $X_i \sim \mathcal{D}_i(x_i|\theta_i) \;\; x_i \in \mathbb{R}$, for each item $i \in S$, and selects i with probability of X_i being the maximum among all the scores of items in S, i.e. $i \sim \mathbf{P}(i|S) = Pr(X_i > X_j \;\; \forall j \in S \setminus \{i\}), \;\; \forall i \in S$

One widely used example of RUM is the *Multinomial-Logit (MNL)* or famously called *Plackett-Luce model (PL)* where $\mathcal{D}_i'$s are independent Gumbel distributions [6], i.e. $\mathcal{D}_i(x_i|\theta_i) = e^{-(x_j - \theta_j)}e^{-e^{-(x_j-\theta_j)}}$. In this case it can be shown that $\mathbf{P}(i|S) = \frac{e^{\theta_{a_i}}}{\sum_{j \in S} e^{\theta_{a_j}}}, \;\; \forall i \in S.$

Similarly, alternative family of discrete choice models can be obtained assuming different distributions over the utility scores X_i, e.g. when $(X_1, \ldots X_n) \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ are jointly normal with mean $\boldsymbol{\theta} = (\theta_1, \ldots \theta_n)$ and covariance $\boldsymbol{\Sigma}$, above reduces to *Multinomial Probit Model (MNP)*, although unlike MNL, choice probabilities $\mathbf{P}(i|S)$ for MNP do not have a closed formed solution [20].

2.2 Measuring performance – Regret

We compare the performance of the learner's strategy with respect to a 'best' arm of the choice model. As defined before, for *pairwise-subset choice models*, the most natural candidate for the 'best' arm is the Condorcet winner $a^* \in [n]$, i.e. $Q_{a^*,a} > \frac{1}{2}, \; \forall a \in [n] \setminus \{a^*\}$, assuming $\mathbf{Q}$ contains a Condorcet arm. Then an intuitive way to define the regret of *Battling-Bandit* is by extending the notion of dueling bandit regret [24, 22, 4] as follows:

$$R_T^{BB} = \sum_{t=1}^{T}\left(\frac{\sum_{a \in S_t}\left(Q_{a^*,a} - \frac{1}{2}\right)}{k}\right), \quad (3)$$

Consequently, the aim of the learner is to play sets S_t at times $t = 1, 2, \ldots$ to keep the regret as *low* as possible which in fact corresponds to playing a^* as many times as possible in S_t, at any round t. Clearly only if the learner plays the set $S_t = (a_1, a_2, \ldots a_k)$ such that $a_i = a^* \; \forall i \in [k]$, the corresponding regret incurred at round t is 0.

Note that, for *linear-subset choice models*, the 'best' arm is $a^* = \text{argmax}_{a \in [n]}\theta_a$, i.e., an arm having the highest utility score, as that happens to be the Condorcet winner of the underlying pairwise preference model $\mathbf{Q}^{\boldsymbol{\theta}}$. Thus using (3), we can similarly define the regret R_T^{BB} in this case as well with $Q^{\boldsymbol{\theta}}_{a^*,a} = \frac{(\theta_{a^*} - \theta_a + 1)}{2}, \forall a \in [n]$.

3 PROPOSED ALGORITHMS

In this section we describe three algorithms for the *Battling-Bandit* problem. The first two algorithms, *Battling-Doubler* and *Battling-MultiSBM*, respectively generalize the two algorithms for utility based dueling bandits (UBDB) Doubler and MultiSBM, proposed by Ailon et al. [4], which essentially address the problem by using classical multi-armed bandit (MAB) algorithm as an underlying black box. Our third algorithm, *Battling-Duel* is based on dueling matches that uses black-box instances of a dueling bandit algorithms for the purpose.

The main advantage of *Battling-Doubler* is that it works even with an infinite set of arms, although its regret guarantee is off by an extra multiplicative factor of $\ln T$. On the other hand, *Battling-MultiSBM* guarantees $O(nk \ln T)$ regret for any finite set of n arms. However both these algorithms are tailored for *linear-subset choice model*, unlike our third algorithm, *Battling-duel*, which in contrast applies to the general class of *pairwise-subset choice models* (Section 2) and is shown to perform optimally with a regret guarantee of $O(n \log T)$ as long as the it uses an optimal dueling bandit algorithm as the black box.

Before describing our proposed algorithms, it is worth describing the black-box algorithms used to design them.

SBM: We call a black box algorithm for the classical

MAB problem[1] as a single bandit machine (SBM). Any SBM instance $\mathcal{S}$ supports three operations: Reset, Advance and Feedback. Reset($\mathcal{S}$) initializes the instance $\mathcal{S}$. Advance($\mathcal{S}$) suggests which arm to play next and Feedback($\mathcal{S}, r$) feedbacks a reward $r \in [0,1]$ to $\mathcal{S}$.

Definition 2. *(**α-robust SBM**) [4] Consider a SBM instance $\mathcal{S}$ with n arms. For any sub-optimal arm $x \in [n]$, let T_x be the number of times x is played by $\mathcal{S}$ in T rounds. The SBM $\mathcal{S}$ is said to be α-robust if $\forall s \geq 4\alpha\Delta_x^{-2}\ln T$, it holds that $\mathbf{P}[T_x > s] < \frac{2}{\alpha}(s/2)^{-\alpha}$, where Δ_x denotes the gap between the expected reward of the best arm and that of arm x in the underlying MAB instance.*

DBM: Similar to SBM, we call a black box algorithm for the dueling bandit problem as dueling bandit machine (DBM). A DBM also supports the same three operations as that of a SBM instance, with the only difference being that a DBM instance $\mathcal{D}$, outputs *two* arms $x, y \in [n]$ on the Advance($\mathcal{D}$) operation instead of one. We refer x as the right arm and y the left arm. Also, in this case, the feedback r upon Feedback($\mathcal{S}, r$) is a preference relation between x and y defined as $r = \mathbf{1}(y \text{ beats } x)$. We now describe our main algorithms and their regret guarantees. Proofs of all the theorems are deferred to the Appendix.

3.1 Battling-Doubler

The first algorithm, *Battling-Doubler*, maintains a single SBM instance $\mathcal{S}$. The total time horizon T is divided into exponentially growing epochs, and a MAB game is played within each epoch using $\mathcal{S}$. Specifically, at any epoch, the algorithm plays the first $(k-1)$ arms uniformly from the multiset of arms $\mathcal{L}$ selected by $\mathcal{S}$ in the previous epoch, the k^{th} arm is played adaptively according to suggestion of the SBM $\mathcal{S}$ upon which $\mathcal{S}$ receives a binary reward based on the defeat or victory of the k^{th} arm it suggested. Algorithm 1 describes *Battling-Doubler* formally.

Remark 3. *Note that when $[n]$ is finite, in order to save the memory overhead of maintaining the multiset $\mathcal{L}$ (line 14), a more elegant approach can be to instead maintain a probability distribution $\mathbf{p}^t \in \Delta_n$ over the n arms, where $p_a^t \in [0,1]$ denotes the fraction of times arm $a \in [n]$ was played as the k^{th} arm of S_{t-1} at round $(t-1)$, and sample $a_1^t, a_2^t, \cdots, a_{k-1}^t$ according to $\mathbf{p}^t$ (in line 7).*

Theorem 3. ***Battling-Doubler Regret for general arm sets.*** *Assume that the SBM $\mathcal{S}$ used by Battling-Doubler has expected regret no more than $c\ln^{\beta}(t)$ at the end of $t \in \mathbb{N}$ rounds, where $c > 0, \beta > 0$ are constants independent of t. Then, under the linear-subset choice model, the expected regret of Battling-Doubler at the end of T rounds is at most $2c\frac{k\beta}{\beta+1}\ln^{\beta+1}(T)$.*

Algorithm 1 Battling-Doubler

```
 1: Initialize: $\mathcal{S} \leftarrow$ an SBM over set of $[n]$ arms
 2:             $\mathcal{L} \leftarrow [n]$
 3:             $\ell \leftarrow 1,\ t \leftarrow 1$
 4: while true do
 5:   reset($\mathcal{S}$)
 6:   for $j = 1, 2, 3 \cdots 2^{\ell}$ do
 7:     Select $a_1^t, a_2^t, \cdots, a_{k-1}^t$ uniformly from $\mathcal{L}$
 8:     $a_k^t \leftarrow$ Advance($\mathcal{S}$)
 9:     Play $S_t = (a_1^t, a_2^t, \cdots a_k^t)$
10:     Receive winner $i_t^* \in [k]$
11:     Feedback($\mathcal{S}, \mathbf{1}(i_t^* = k)$)
12:     $t \leftarrow t + 1$
13:   end for
14:   $\mathcal{L} \leftarrow$ the multiset of arms played as $a_k^t$ in epoch $\ell$
15:   $\ell \leftarrow \ell + 1$
16: end while
```

Corollary 4. ***Battling-Doubler Regret for finite set of arms.*** *Assume the SBM $\mathcal{S}$ used in Battling-Doubler is the Upper Confidence Bound (UCB) algorithm [5] and suppose the underlying feedback model used for the* Battling-Bandit *problem is* linear-subset choice model *with parameter $\boldsymbol{\theta} \in [0,1]^n$, such that $\theta_1 > \max_{i=2}^{n}\theta_i$. Then the expected regret of Battling-Doubler is $O(kH\log^2 T)$, where $H := \sum_{i=2}^{n}\frac{1}{\Delta_i}$, and $\Delta_i = \theta_1 - \theta_i\ \forall i \in [n]$.*

Note that, the above regret guarantee becomes trivial if the gap parameter H is large. Instead, we can also derive the a gap-independent regret bound as follows:

Corollary 5. ***Battling-Doubler Regret (Gap-independent regret bound)*** *Assume that the SBM $\mathcal{S}$ in Battling-Doubler is the Upper Confidence Bound (UCB) algorithm [5]. Then under any linear-subset choice model, the expected regret of Battling-Doubler is at most $O(k\sqrt{nT\log^3 T})$.*

3.2 The Battling-MultiSBM algorithm

Unlike *Battling-Doubler*, *Battling-MultiSBM* simultaneously maintains n independent SBMs $\mathcal{S}_a$, $\forall a \in [n]$. At each round t, the first $(k-1)$ arms are played according to the last $(k-1)$ arms of round $(t-1)$ and the k^{th} arm is played according to the suggestion of SBM $\mathcal{S}_{a_k^{t-1}}$ which corresponds to the k^{th} arm played at round $t-1$. As before, a binary reward is fed back to $\mathcal{S}_{a_k^{t-1}}$ based on whether the arm it suggested at round t wins or not. *Battling-MultiSBM* is formally described in Algorithm 2.

Theorem 6. ***Battling-MultiSBM Regret with finite arms.*** *Suppose all the SBMs used*

[1] Given a fixed set of n arms, each associated to a reward distribution with their expectation bounded in the range $[0,1]$, the classical MAB defines the problem of identifying the best arm with highest expected reward by actively selecting one arm at each round sequentially and receiving a feedback from its underlying reward distribution in an online fashion [5].

Algorithm 2 Battling-MultiSBM

```
Initialize: For each arm $a \in [n], \mathcal{S}_a \leftarrow$ new SBM over set of arms $[n]$. Reset($\mathcal{S}_a$).
Select $a_2^0, a_3^0, \cdots a_k^0$ uniformly from $[n]$
for $t = 1, 2, \cdots T$ do
    $a_j^t = a_{j+1}^{t-1}, \forall j \in [k-1]$
    $a_k^t \leftarrow$ advance$\left(\mathcal{S}_{a_{k-1}^t}\right)$
    Play $S_t = (a_1^t, a_2^t, \cdots a_k^t)$
    Receive winner $i_t^* \in [k]$
    Feedback $\left(\mathcal{S}_{a_{k-1}^t}, \mathbf{1}(i_t^* = k)\right)$
end for
```

Algorithm 3 Battling-Duel

```
Initialize: $\mathcal{D} \leftarrow$ new dueling bandit algorithm over set of $[n]$ arms
for $t = 1, 2, \cdots$ do
    $\{x_t, y_t\} \leftarrow$ Advance($\mathcal{D}$)
    $S_t$=$(x_t, \ldots, x_t, y_t, \ldots, y_t)$, where $x_t$ and $y_t$ are respectively replicated for $\lfloor k/2 \rfloor$ and $\lceil k/2 \rceil$ or $\lceil k/2 \rceil$) and $\lfloor k/2 \rfloor$ times, each with probability $\frac{1}{2}$.
    Receive winner $i_t^* \in [k]$
    Feedback: $(\mathcal{D}, \mathbf{1}(S_t(i_t^*) = y_t))$
end for
```

in Battling-MultiSBM are α-robust, where $\alpha = \max\{3, 2 + \frac{\ln K}{\ln \ln T}\}$. Also assume $\theta_1 > \max_{i=2}^n \theta_i$, $\Delta_i = (\theta_1 - \theta_i)$, $\forall i \in [n]$ and $H := \sum_{i=2}^n \frac{1}{\Delta_i}$. Then, under the linear-subset choice model with parameter $\boldsymbol{\theta} \in [0,1]^n$, the regret of Battling-MultiSBM is $O\left(kH\alpha\left(\ln T + n\ln n + n\ln\ln T + 2\sum_{i=2}^n \ln\frac{1}{\theta_1 - \theta_i}\right)\right)$.

Note that the above bound is essentially of $O(nk\log T)$, since $H = O(n)$ given a fixed instance of *linear-subset choice model* $\boldsymbol{\theta}$. Similar *Battling-Doubler*, here also we can derive a gap-independent regret bound as follows:

Corollary 7. ***Battling-MultiSBM Regret (Gap independent regret bound).*** *If the SBMs used in Battling-MultiSBM are α-robust, $\alpha = \max\{3, 2 + \frac{\ln K}{\ln \ln T}\}$, then under any linear subset choice model, the regret of Battling-MultiSBM is $O\left(k\sqrt{nT}\alpha\left(\sqrt{\ln T} + n\frac{(\ln n + \ln \ln T)}{\sqrt{\ln T}}\right)\right)$.*

3.3 Battling-Duel

Our third algorithm *Battling-Duel*, is a simple general algorithm for *Battling-Bandits* that uses a good (low-regret) dueling bandit algorithm as its black-box and works under any *pairwise-subset choice model*. *Battling-Duel* maintains an instance of a dueling bandit algorithm (DBM) $\mathcal{D}$, at each round t, the algorithm receives two arms $x_t, y_t \in [n]$ from $\mathcal{D}$, and plays the multiset $S_t = (x_t, x_t, \ldots, x_t, y_t, y_t, \ldots, y_t)$ of k arms by replicating x_t and y_t equal number of times on an average. More precisely, x_t is replicated for either $\lfloor k/2 \rfloor$ or $\lceil k/2 \rceil$ number of times with equal probability of $\frac{1}{2}$ and the rest half of S_t is filled with y_t. Upon playing S_t, once the identity of the battling winner is revealed, $\mathcal{D}$ receives a corresponding dueling feedback depending on if its x_t or y_t. The formal description is given in Algorithm 3.

The following result shows an exact equivalence between the regret of *Battling-Duel* $R_T^{BB}(BD)$ and that of its underlying dueling bandit algorithm $R_T^{DB}(\mathcal{D})$.

Theorem 8. ***Battling-Duel Regret.*** *Under any pairwise-subset choice model with preference matrix $\mathbf{Q}$, the regret incurred by Battling-Duel (BD) in T rounds is*

$$R_T^{BB}(BD) = \kappa R_T^{DB}(\mathcal{D}),$$

where $\kappa = \frac{2(k-1)}{k}$ if k is even, or $\kappa = \frac{2k}{k+1}$ otherwise. $R_T^{DB}(\mathcal{D})$ is the regret incurred by $\mathcal{D}$ in T rounds, i.e. $R_T^{DB}(\mathcal{D}) = \sum_{t=1}^T \frac{(Q'_{a^*,x_t} - \frac{1}{2}) + (Q'_{a^*,y_t} - \frac{1}{2})}{2}$ as per the standard definition of regret for any dueling bandit algorithm $\mathcal{D}$ [24, 22] under $\mathbf{Q}'$ (as also obtained from (3) *with $k = 2$), $\mathbf{Q}'$ being the pairwise preference model perceived by $\mathcal{D}$ in Algorithm 3, such that $Q'_{x_t,y_t} := \mathbf{P}(i_t^* == x_t)$, for any choices of (x_t, y_t), at any round t.*

Using $\mathcal{D}$ as the state-of-the-art RUCB algorithm [24], gives the following regret guarantee for *Battling-Doubler*:

Corollary 9. ***Battling-Duel Regret with RUCB.*** *Assume that the DBM $\mathcal{D}$ in Battling-Duel is RUCB [24], then under any pairwise-subset choice models with preference matrix $\mathbf{Q}$, the regret of Battling-Duel is*

$$\kappa\left(\tilde{C} + \sum_{i \in [a] \setminus \{a^*\}} \frac{2\alpha(\Delta_i + 4\Delta_{\max})}{\Delta_i^2} \ln T\right), \qquad (4)$$

where $\tilde{C}$ is a problem instance (i.e. $\mathbf{Q}$) dependent constant, independent of the time horizon T, $\Delta_i = \left(Q_{a^*,i} - \frac{1}{2}\right)$, $\Delta_{\max} = \max_{i \in [n]} \Delta_i$, $\forall i \in [n]$, $\kappa = \frac{2(k-1)}{k}$ if k is even, or $\kappa = \frac{2k}{k+1}$ otherwise.

Note that Corollary 9 essentially gives an $O(n \log T)$ regret guarantee for *Battling-Duel* since the first term of (4) is constant given a fixed $\mathbf{Q}$, whereas the second term scales as $\log T$ for each $(n-1)$ suboptimal arms. Clearly *Battling-Duel* performs the best in terms of dependency of its regret guarantee on n, k and T, among all three of our proposed algorithms. We next establish a matching regret lower bound of $\Omega(n \log T)$ for the problem, which essentially proves the optimality of *Battling-Duel*.

4 REGRET LOWER BOUND

In this section, we derive an $\Omega(n \ln T)$ regret lower bound (Theorem 8) for the problem of *Battling-Bandit* under *any* *pairwise-subset choice model*. Our proof involves reduction of an instance of the *Dueling-Bandit (DB)* problem to an instance of the *Battling-Bandit (BB)* problem and solve the former using an algorithm designed for the later. More specifically, we first prove the following key result:

Theorem 10 (Reducing *Dueling-Bandit* to *Battling-Bandit*)**.** *There exists a reduction from the Dueling-Bandits problem to Battling-Bandits, which preserves expected regret under any pairwise-subset choice model.*

Proof. Consider that we have an algorithm $\mathcal{A}_{BB}$ for the BB problem and our goal is to construct a DB algorithm $\mathcal{A}_{DB}$ using this. One intuitive way to do this is: At any round t, first play $\mathcal{A}_{BB}$ to generate the set S_t of k arms, randomly sample two indices $i_t, j_t \in [k]$ from the set $[k]$, play a_{i_t}, a_{j_t} respectively as the left and right arm of DB, receive the winner w_t of the duel (a_{i_t}, a_{j_t}) from the DB environment and feedback a winning index i_t^* to $\mathcal{A}_{BB}$ accordingly as the winner of the S_t battle. The resulting algorithm $\mathcal{A}_{DB}$ is as summarized in Algorithm 4.

Algorithm 4 $\mathcal{A}_{DB}$**: Reducing *DB* to *BB***

```
for $t = 1, 2, \ldots$ do
    $S_t \leftarrow$ Multiset of arms played by $\mathcal{A}_{BB}$ at round $t$
    Draw $i_t, j_t \sim$ Unif$[k]$ (without replacement)
    Play $(a_{i_t}, a_{j_t})$
    Receive feedback $w_t = \mathbf{1}(\{a_{i_t}$ beats $a_{j_t}\})$
    Return $i_t^* = i_t w_t + j_t(1 - w_t) \in \{i_t, j_t\}$ to $\mathcal{A}_{BB}$
        as winning index to $\mathcal{A}_{BB}$
end for
```

The crucial observation is that if the DB environment actually simulates the winner from an underlying (unknown) preference matrix $\mathbf{Q}$, then internally $\mathcal{A}_{BB}$ sees a world where subset choice probabilities are given by $\mathbf{P}(i|S) = \frac{\sum_{j=1, j\neq i}^{k} 2Q_{a_i, a_j}}{k(k-1)}$, due to Lemma 1. Thus at each round t, the average instantaneous regret of $\mathcal{A}_{DB}$ is:

$$\begin{aligned}
&\mathbf{E}_{i_t, j_t \sim [k], i_t \neq j_t}[r_t(\mathcal{A}_{DB})] \\
&= \mathbf{E}_{i_t, j_t \sim [k], i_t \neq j_t}\left[\frac{(Q_{a^*, a_{i_t}} - \frac{1}{2}) + (Q_{a^*, a_{j_t}} - \frac{1}{2})}{2}\right] \\
&= \frac{1}{k(k-1)} \sum_{i=1}^{k} 2(k-1)\left[\frac{(Q_{a^*, i} - \frac{1}{2})}{2}\right] \\
&= \sum_{i=1}^{k}\left[\frac{(Q_{a^*, i} - \frac{1}{2})}{k}\right] = r_t(\mathcal{A}_{BB}).
\end{aligned}$$

where the second equality follows since the expectation is taken over the random draw of two indices i_t, j_t from $[k]$ without replacement, $a^* \in [n]$ being the Condorcet arm of $\mathbf{Q}$ and $r_t(\mathcal{A}_{BB})$ denotes the instantaneous regret of $\mathcal{A}_{BB}$ at round t, as defined in Section 2.2. Thus we get $\mathbf{E}[R_T(\mathcal{A}_{DB})] = \mathbf{E}\left[\sum_{t=1}^{T} r_t(\mathcal{A}_{DB})\right] = \sum_{t=1}^{T} r_t(\mathcal{A}_{BB}) = R_T(\mathcal{A}_{BB})$, proving the claim. □

Corollary 11. *Given any algorithm $\mathcal{A}_{BB}$ for Battling-Bandits (BB) under pairwise-subset choice model associated to a preference matrix $\mathbf{Q}$ with Condorcet winner, there exists a problem instance of BB such that*

$$\liminf_{T\to\infty} \frac{\mathbf{E}[R_T(\mathcal{A}_{BB})]}{\ln T} \geq \sum_{i\in[n]\setminus\{a^*\}} \min_{j\in B_i} \frac{\Delta_{ij}}{kl(Q_{i,j}, \frac{1}{2})},$$

where $\Delta_{ij} = \frac{(Q_{a^*,i} - \frac{1}{2}) + (Q_{a^*,j} - \frac{1}{2})}{2}$, $B_i = \{j \mid Q_{i,j} < \frac{1}{2}\}$, and $kl(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ denotes the kl-divergence between two Bernoulli distributions with parameters p and q.

Remark 4. Note that Corollary 11 implies that the asymptotic regret lower bound is $\Omega(n\log T)$ since $\sum_{i\in[n]\setminus\{a^*\}} \min_{j\in B_i} \frac{\Delta_{ij}}{\text{kl}(Q_{ij}, \frac{1}{2})}$ essentially involves a sum over $(n-1)$ terms, each being a constant for a fixed $\mathbf{Q}$, thus making it $\Omega(n)$. This therefore concludes that the regret guarantee of *Battling-Duel* (Theorem 8) is indeed optimal when used with a 'good' dueling bandit algorithm of $O(n\log T)$ regret guarantee (Corollary 9).

Remark 5. The optimal regret guarantee of $O(n\log T)$ of *Battling-Bandits* with *pairwise-subset choice model* is independent of the subset size k, which essentially clarifies the *tradeoff of learning rate with subset size k — even with the flexibility of playing larger k-sized sets* $(k \geq 2)$ *does not help in faster information aggregation than the corresponding dueling setup* $(k = 2)$ *— which might appear counter intuitive but is justified as information theoretically the winner information of a k-set does not reveal any additional information over that in a 2-set.*

5 EXPERIMENTS

We now present empirical evaluations for our proposed algorithms on different synthetic and real world datasets and also compare them with the *Self-Sparring* algorithm of [17], which is the only existing work applicable to our framework. In all our experimental results, our proposed algorithm *Battling-Duel* outperforms the rest, rightfully justifying the optimality of its regret guarantees as discussed in Remark 4. A detailed discussion is given below:

Algorithms. We compared the performances of the following 5 algorithms: **1. BD-RUCB**: *Battling-Duel* (Section 3.3) with *RUCB* [24] as the DBM $\mathcal{D}$. **2. BD-TS**: *Battling-Duel* (Section 3.3) with *Double-Thompson Sampling* [21] as the DBM $\mathcal{D}$. **3. B-Dblr**: *Battling-Doubler* (Section 3.1) with *UCB* [5] as the SBM $\mathcal{S}$. **4. B-Msbm**: *Battling-MultiSBM* (Section 3.2) with *UCB* [5] as the

SBM $\mathcal{S}$. **5. SS-TS**: *Self-Sparring* algorithm [17] with Thompson Sampling [3]. This algorithm closely resembles to the Sparring algorithm of [4] that maintains a single copy of SBM (a MAB algorithm), and at each round t, it queries the SBM k times to produce a k-sized battling set S_t. To the best of our knowledge no other existing work applies to the setup of *Battling-Bandits*.

Experimental Setup and Performance Measures We plot regret of each of the 5 algorithms for different real world and synthetic datasets, as describe in Section 5.1 and 5.2. In all the experiments the time horizon is fixed to $T = 5000$ (with few exceptions if the regret plot do not converge within 5000 time iterations) and the experiments are run for different item sizes n and subset sizes k as specified in the corresponding experiments. The measure of performances in all the plots is the total regret R_T^{BB} in T round as defined in (3). All results are reported as average across 50 runs along with the standard deviations.

5.1 Experiments on Synthetic Datasets

For synthetic experiments with *linear-subset choice model* (Section 2), we use the following four different utility score vectors $\boldsymbol{\theta}$: 1. *arith* 2. *geom* 3. *g1* and 4. *g3*.

Both *arith* and *geom* has $n = 8$ items, with item 1 being the 'best' (Condorcet) item of highest score, i.e. $\theta_1 > \max_{i=2}^{8} \theta_i$; the rest of the θ_is are in an arithmetic or geometric progression respectively, as their names suggest. The two score vectors are described in Table 2.

arith	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
geom	0.8	0.7	0.512	0.374	0.274	0.2	0.147	0.108

Table 1: Parameters for *linear-subset choice model*

The next two utility score vectors has $n = 15$ items in each. Similarly as before, item 1 is the Condorcet winner here as well, with $\theta_1 > \max_{i=2}^{8} \theta_i$. More specifically for *g1*, the individual score vectors are of the form: $\theta_i = 0.8, \quad \text{if } i = 1$ and $\theta_i = 0.2, \ \forall i \in [15] \setminus \{1\}$. For *g3*, the individual score vectors are of the form: $\theta_i = 0.8, \quad \text{if } i = 1, \theta_i = 0.7, \ \forall i \in [8] \setminus \{1\}$ and $\theta_i = 0.6, \text{ otherwise}$

Clearly, *g3* is a harder model (for learning the Cordorcet item), than *g1* as in the former case, the gap between the items scores are very close to each other and the best and the second best item is only 0.1 distance apart, whereas gap is 0.6 for every suboptimal items in the later case. The fact is reflected in our experimental results as well.

Results on linear-subset choice model. Figure 1 shows the comparative regret performances of the 5 algorithms, for Battling-Bandits with *linear-subset choice model* on 4 different utility score vectors as described above. We set $k = 4$ for *arith* and *geom* and $k = 8$ for the rest two. The results clearly shows the superiority of *Battling-Duel* compared to the rest. In fact, BD-TS performs slightly better than BD-RUCB as Thompson sampling based algorithms are known to perform empirically well compared to UCB based algorithms (in spite of both1 having a similar $O(n \log T)$ regret guarantee), although it comes at the cost of a higher performance variability as evident from our plots. SS-TS being a Thompson Sampling based algorithm, it exhibits a very high variability too.

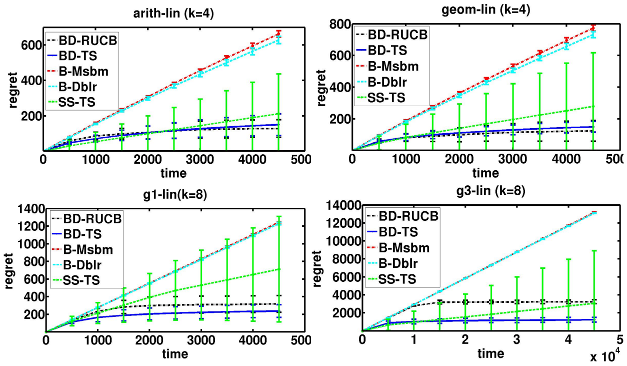

Figure 1: Averaged regret over time on synthetic datasets (on *linear-subset choice model*)

Results on MNL choice model. We also run the above experiment for the same 4 utility scores 1. *arith* 2. *geom* 3. *g1* and 4. *g3* on *Multinomial Logit (MNL)* choice model (as describes in Section 2). Similarly as before, even in this case the two *Battling-Duel* algorithms, BD-RUCB and BD-TS, perform the best among all 5. As argued before, *g3* being the "hardest instance to learn", for both *linear* and MNL choice models, we had to run the algorithms for comparatively larger number of iterations until convergence. The results are shown in Figure 2.

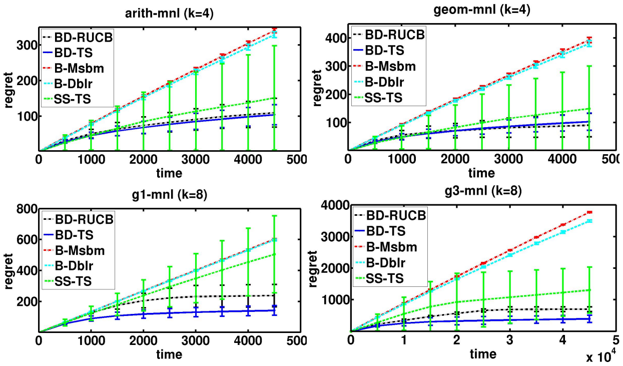

Figure 2: Averaged regret over time on synthetic datasets (on *multinomial logit (MNL) model*)

Results on Pairwise-subset choice model. We finally run experiments for the general *pairwise-subset choice model* on two synthetic pairwise preference matrices: *arxiv-pref* and *arith-pref* with $n = 6$ and $n = 8$ respectively. See Appendix E.2 for the details of the datasets.

We run the experiments for $k = 4$ for both the datasets. As before, the two *Battling-Duel* algorithms excel the rest in this case as well, as follows from Figure 3.

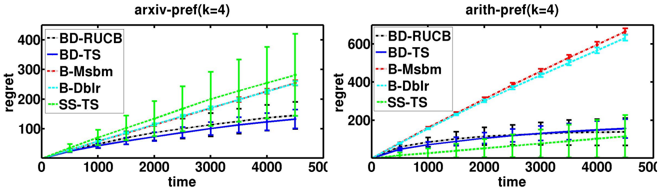

Figure 3: Averaged regret over time on synthetic datasets (on *pairwise-subset choice model*)

5.2 Experiments on Real Datasets

We also evaluated our method on four real-world preference learning datasets: 1. *Car* [1] 2. *Hurdy* [16] 3. *Tennis* [16] and 4. *Sushi* [11]. Each of the dataset contains pairwise preferences of a given set of n items, where $n = 10$ for both *Car* and *Hurdy*, and it is respectively 8 and 16 for *Tennis* and *Sushi*. All the preference matrices contain a Condorcet winner (as required as per our problem setup in Section 2). We set $k = 6$ for both *Hurdy* and *Tennis* and respectively 4 and 10 for *Car* and *Sushi*. The description of the datasets along with data extraction procedure and the actual preference matrices are given in Appendix E.

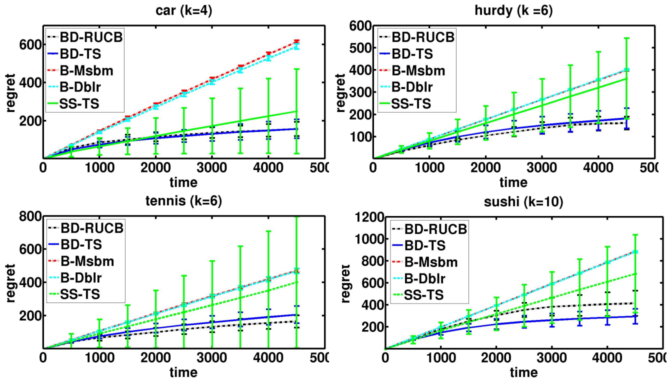

Figure 4: Averaged regret over time on real datasets (on *pairwise-subset choice model*)

Results. Figure 4 shows the comparative regret performances of the 5 algorithms used. As expected, BD-TS turns out to be the best algorithm for most of the cases, with BD-RUCB following it closely, whereas B-BMsbm and B-Dblr performs poorly in comparison, rightfully justifying their suboptimal regret guarantees (Theorem 3 and 6). SS-TS shows a very high variability as usual and performs worse than both BD-RUCB and BD-TS.

5.3 Effect of varying subset size k

We also analyze the scaling of the regret performances our optimal algorithm *Battling-Duel* with increasing k. We use BD-RUCB for the purpose on two score vectors 1. *g1* and 2. *g1-big* with varying k, keeping n fixed to 15 and 50 respectively. Here *g1-big* is just a larger version of *g1* utility score with $n = 50$ items, such that $\theta_1 = 0.8$ and $\theta_i = 0.2, \ \forall i \in [50] \setminus \{1\}$ (see Appendix E for details). The results are shown in Figure 5, which clearly reflects that the learning rate of *Battling-Duel* does not scale with k, justifying that its regret guarantee is indeed independent of the subset size k (Theorem 8).

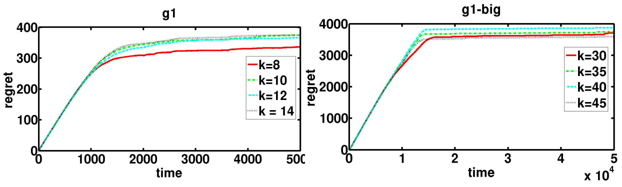

Figure 5: Averaged regret over time with varying k and fixed n (on *linear-subset choice model*)

6 CONCLUSION AND FUTURE WORK

We introduce the problem of *Battling-Bandit* – generalization of the well-studied *Dueling-Bandit* problem, where the objective is to find the ‘best’ arm by successively playing a subset of k arms from a pool of n arms and subsequently receiving the winner feedback in an online fashion. For this we develop a novel k-wise feedback model, viz. *pairwise-subset choice model* and propose three algorithms along with their regret bound guarantees. We also show a matching regret lower bound of $\Omega(n \log T)$ proving the optimality of our algorithms.

Our proposed framework of *Battling-Bandits* opens up a set of new directions to explore – with different choices of feedback models, regret objectives, or even applying this to new settings like revenue maximization, contextual or adversarial bandits etc. One very interesting point noted here is that the optimal regret guarantee is independent of the subset size $k \geq 2$, which implies the flexibility of playing larger subsets does not really help to gather information faster than the corresponding dueling case ($k = 2$), atleast with the current *pairwise-subset choice feedback model*. It will be interesting to study the tradeoff of the subset size on the regret (learning rate to identify the ‘best’ arm) for different subset choice models, e.g. MNL, MNP etc. Lastly, it would also be useful to analyze other dueling bandit algorithms, e.g. Sparring [4], especially for large set of structured arms and their implications in solving *Battling-Bandit* with different settings.

ACKNOWLEDGEMENTS

Thanks to Arun Rajkumar (Conduent Labs, India) for all the useful discussions and the anonymous reviewers for their insightful comments. This work is supported by Qualcomm, ACM-India and IARCS travel grant.

References

[1] Ehsan Abbasnejad, Scott Sanner, Edwin V Bonilla, Pascal Poupart, et al. Learning community-based preferences via dirichlet process mixtures of gaussian processes. In *IJCAI*, pages 1213–1219, 2013.

[2] Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. A near-optimal exploration-exploitation approach for assortment selection. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 2016.

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

[4] Nir Ailon, Zohar Shay Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *ICML*, volume 32, pages 856–864, 2014.

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

[6] Hossein Azari, David Parks, and Lirong Xia. Random utility theory for social choice. In *Advances in Neural Information Processing Systems*, pages 126–134, 2012.

[7] Gerardo Berbeglia and Gwenaël Joret. Assortment optimisation under a general discrete choice model: A tight analysis of revenue-ordered assortments. *arXiv preprint arXiv:1606.01371*, 2016.

[8] Brian Brost, Yevgeny Seldin, Ingemar J. Cox, and Christina Lioma. Multi-dueling bandits and their application to online ranker evaluation. *CoRR*, abs/1608.06253, 2016.

[9] Antoine Désir, Vineet Goyal, Srikanth Jagabathula, and Danny Segev. Assortment optimization under the mallows model. In *Advances in Neural Information Processing Systems*, pages 4700–4708, 2016.

[10] Antoine Désir, Vineet Goyal, Danny Segev, and Chun Ye. Capacity constrained assortment optimization under the markov chain based choice model. *Operations Research*, 2016.

[11] Toshihiro Kamishima and Shotaro Akaho. Efficient clustering for orders. In *Mining Complex Data*. Springer, 2009.

[12] Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Dcm bandits: Learning to rank with multiple clicks. In *International Conference on Machine Learning*, pages 1215–1224, 2016.

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

[14] Rémi Munos et al. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

[15] Kameng Nip, Zhenbo Wang, and Zizhuo Wang. Assortment optimization under a single transition model. 2017.

[16] Siddartha Y Ramamohan, Arun Rajkumar, and Shivani Agarwal. Dueling bandits: Beyond condorcet winners to general tournament solutions. In *Advances in Neural Information Processing Systems*, pages 1253–1261, 2016.

[17] Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. *arXiv preprint arXiv:1705.00253*, 2017.

[18] Csaba Szepesvari and Tor Lattimore. *Bandit Algorithms*, 2016. http://banditalgs.com/2016/09/18/the-upper-confidence-bound-algorithm/.

[19] Kalyan Talluri and Garrett Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 2004.

[20] Ondrej Vojacek, Iva Pecakova, et al. Comparison of discrete choice models for economic environmental research. *Prague Economic Papers*, 2010.

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

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

[23] Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten De Rijke. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pages 307–315, 2015.

[24] Masrour Zoghi, Shimon Whiteson, Remi Munos, Maarten de Rijke, et al. Relative upper confidence bound for the k-armed dueling bandit problem. In *JMLR Workshop and Conference Proceedings*, number 32, pages 10–18. JMLR, 2014.

[25] Masrour Zoghi, Shimon A Whiteson, Maarten De Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 73–82. ACM, 2014.