Dueling Bandits with Qualitative Feedback

Liyuan Xu,[1,2] **Junya Honda,**[1,2] **Masashi Sugiyama**[1,2]
[1]The University of Tokyo, [2]RIKEN
liyuan@ms.k.u-tokyo.ac.jp, honda@stat.t.u-tokyo.ac.jp, sugi@k.u-tokyo.ac.jp

Abstract

We formulate and study a novel multi-armed bandit problem called the *qualitative dueling bandit (QDB)* problem, where an agent observes not numeric but qualitative feedback by pulling each arm. We employ the same regret as the *dueling bandit* (DB) problem where the duel is carried out by comparing the qualitative feedback. Although we can naively use classic DB algorithms for solving the QDB problem, this reduction significantly worsens the performance—actually, in the QDB problem, the probability that one arm wins the duel over another arm can be *directly* estimated without carrying out actual duels. In this paper[1], we propose such direct algorithms for the QDB problem. Our theoretical analysis shows that the proposed algorithms significantly outperform DB algorithms by incorporating the qualitative feedback, and experimental results also demonstrate vast improvement over the existing DB algorithms.

1 Introduction

The stochastic multi-armed bandit (MAB) problem is a sequential decision-making problem, where an agent repeatedly chooses one option from K alternatives (which are often called arms). At each round, the agent receives a random reward that depends on the arm being selected, and the goal is to maximize the cumulative reward. This problem has been extensively studied for many years, both from theoretical and practical aspects. Numerous algorithms have been proposed for the problem (Thompson 1933; Auer 2003) and applied to various fields including the design of clinical trials (Villar, Bowden, and Wason 2015), economics (Rothschild 1974) and crowdsourcing (Zhou, Chen, and Li 2014).

The dueling bandit (DB) problem (Yue et al. 2012) is a variant of the MAB problem, where an agent only observes the result of a "duel", a noisy comparison between the selected two arms. While the MAB problem assumes that the feedback is numeric, the DB problem only assumes that the arms are comparable based on the feedback. Therefore, it is useful when the numeric feedback is not available, such as information retrieval and clinical trials.

[1]The longer version including all appendices is available at https://arxiv.org/abs/1809.05274

Even when the numeric feedback is not available, we may still have direct access to qualitative feedback. For example, in information retrieval, users might report the relevance of a document returned by a system on a scale of "Irrelevant"—"Partially Relevant"—"Relevant". In such a situation, we can consider a special kind of the DB problem first introduced by Busa-Fekete et al. (2013), which we call the *qualitative DB (QDB)* problem.

In the QDB problem, an agent pulls one arm at each round and observes qualitative feedback. Although a duel is not conducted explicitly in the QDB problem, we consider algorithms to minimize the same regret as the DB problem, in which the probability of an arm winning a duel with another arm corresponds to the probability of the arm getting higher qualitative feedback than the other. Therefore, we can adapt any algorithms for the DB problem to the QDB problem by converting the feedback in every two rounds into the result of one duel.

However, this reduction significantly worsens the performance because, in the QDB problem, the winning probability can be calculated from the estimated feedback distributions. Busa-Fekete et al. (2013) also partially considered this problem, and they improved the performance of the classic DB algorithms by constructing a tight confidence bound. However, they still use the same exploration strategy as the classic DB algorithm. In this paper, we show that we can further improve the performance by designing a specific exploration strategy for the QDB problem.

Several definitions of the "best arm" have been proposed for the DB problem. In this paper, we consider two types of winners, the *Condorcet winner* and the *Borda winner*, both of which are defined in Section 3, and we propose algorithms for each winner. The proposed algorithms are inspired by algorithms in the MAB, namely Thompson sampling (Thompson 1933) and the upper confidence bound (UCB) algorithm (Auer 2003). Interestingly, the algorithm based on Thompson sampling, one of the most popular algorithms for the MAB problem, only works for the criterion of the Condorcet winner and suffers polynomial regret in a specific instance in the criterion of the Borda winner.

The rest of paper is structured as follows. After discussing related work in Section 2, we formulate the QDB problem in detail in Section 3. We introduce two formulations of the QDB problem and propose algorithms for these problems in

Sections 4 and 5. Lastly, we show empirical results for an information retrieval setting in Section 6.

2 Related Work

There are two lines of researches that are related with the QDB problem. The first is the DB problem (Yue et al. 2012), which is the MAB problem with feedback given as a form of noisy comparison between two arms. Many researches have been conducted for this problem and some of them discuss specific comparison models. For example, Hofmann, Whiteson, and de Rijke (2011) have discussed the case where a duel is carried out by interleaved comparison with some user model, and Yue et al. (2012) introduced the Bradley-Terry model. Among them, several models involve random variables corresponding to the utilities associated with arms, and the result of a duel is determined by the order of such variables. For example, a Gaussian model (Yue et al. 2012) is the case where the random variables follows a Gaussian distribution, and Busa-Fekete et al. (2013) considered the case where random variables are on a partially ordered set in the QDB problem.

In the DB problem, the definition of the "best arm" is no longer straightforward because there may exist cyclic preference. Although early work of the DB assumes the total order on arms to ensure the existence of the maximal element, recent work has mainly sought to design algorithms for finding the *Condorcet winner* (Urvoy et al. 2013), which is the arm that wins over all the other arms with probability larger than or equal to $1/2$. This definition can be regarded as a natural generalization of the maximal element, since the Condorcet winner coincides with the maximal element when the total order exists. A number of algorithms have been proposed for the Condorcet winner, for example Urvoy et al.; Komiyama et al.; Wu and Liu (2013; 2015; 2016).

A drawback of this formulation is that the Condorcet winner does not always exist. In such cases, we may introduce other notions of the winners, such as the *Borda winner* (Urvoy et al. 2013) and the *Copeland set* (Zoghi et al. 2015). Ramamohan, Rajkumar, and Agarwal (2016) introduced numerous notions of the winners other than the Condorcet winner.

The other line of related work is the qualitative multi-armed bandit (QMAB) problem (Szorenyi et al. 2015), in which an agent also receives qualitative feedback according to the chosen arm. The difference between the QDB problem and the QMAB problem is that the QDB problem handles winners defined in the classic DB problem, while the QMAB problem introduces its own definition of a "winner", i.e., the arm with the highest τ-quantile of the feedback distribution for $\tau \in (0, 1)$.

This definition is, however, sometimes problematic since it ignores the difference in the feedback distribution below the τ-quantile. Let us consider the case where we have two types of medicines, A and B, and want to figure out which has less side effect. To this end, we may perform clinical trials and obtain feedback from patients about the severeness of side effects.

Table 1: An instance that requires a careful choice of τ in the QMAB problem.

	No side effect	Moderate	Severe
Medicine A	0.995	0.003	0.002
Medicine B	0.995	0.002	0.003

Assume that the feedback is reported on the scale of "No side effect"—"Moderate"—"Severe" and the true probabilities of getting each feedback are shown in Table 1. Then, we can clearly conclude that medicine A is more preferable since it has a less probability of having a severe side effect, and in fact, medicine A becomes the winner in the formulation of the QDB problem. However, the QMAB problem regards these medicines equally good unless $\tau \leq 0.005$ since the τ-quantile feedback is the same. Nevertheless, setting $\tau \leq 0.005$ is almost impossible in practice since we do not have access to the true probabilities beforehand.

On the other hand, the definitions of winners considered in the QDB problem are well-studied in the context of voting theory (see Charon and Hudry (2010), for a survey), and they do not have any hyper-parameter to define the problem itself. This makes our algorithms more applicable to real-world problems.

3 Problem Formulation

We formulate the QDB problem in this section. As in the MAB problem, we consider K arms associated with feedback distributions $\nu_1, \ldots, \nu_K$, and at each round t, the agent chooses one arm $a_t \in [K] = \{1, \ldots, K\}$ and receives feedback r_t sampled from distribution ν_{a_t}. While the MAB problem assumes $\{\nu_i\}$ to be distributions on real values, the QDB considers qualitative feedback which corresponds to the case where $\{\nu_i\}$ are the distributions on the totally ordered set $(\mathcal{L}, \preceq)$, where $\mathcal{L}$ is the set of possible feedback and $\preceq$ denotes a total order between feedback. For simplicity, we assume that $\mathcal{L} = [L]$ and total order $\preceq$ corresponds to order relation $\leq$, which means $1 \preceq 2 \cdots \preceq L$. Thus, distributions $\{\nu_i\}_{i=1}^K$ are all categorical, supports of which are $[L]$. Note that even though the rewards r_t are nominal for notational simplicity, the sum of the feedback has no meaning in the QDB setting.

The QDB problem aims to minimize the same regret as the classic DB problem, which is defined based on pairwise comparison. Following early work (Busa-Fekete et al. 2013), we characterize $\mu_{i,j}$, the probability of arm i winning over arm j, as

$$\mu_{i,j} = \mathbb{P}\left[X_i > X_j\right] + \frac{1}{2}\mathbb{P}\left[X_i = X_j\right],$$

where X_i and X_j are mutually independent random variables following distributions ν_i and ν_j, respectively.

We consider two types of winners in this paper. The first one is the Condorcet winner, which is the arm that wins all the other arms with probability larger than or equal to $1/2$. Formally, arm i^* is the Condorcet winner if $\mu_{i^*,j} \geq 1/2$ for all $j \neq i^*$. We denote the Condorcet winner as a^*_{CW}, and the goal of the QDB problem when employing the Condorcet

winner is to minimize the following regret:

$$R_T^{\mathrm{CW}} = \sum_{t=1}^{T} \Delta_{a_t}^{\mathrm{CW}},$$

where $\Delta_i^{\mathrm{CW}} = \mu_{a^*_{\mathrm{CW}},i} - 1/2$.

The second winner is the Borda winner, which is the arm with the largest Borda score, the average of the winning probabilities against other arms. Formally, the Borda score B_i for arm i is defined as

$$B_i = \frac{1}{K-1} \sum_{j \neq i} \mu_{i,j},$$

and thus, the Borda winner a^*_{BW} is defined as $a^*_{\mathrm{BW}} = \arg\max_{i \in [K]} B_i$. The regret to minimize in this case is formulated as

$$R_T^{\mathrm{BW}} = \sum_{t=1}^{T} \Delta_{a_t}^{\mathrm{BW}},$$

where $\Delta_i^{\mathrm{BW}} = B_{a^*_{\mathrm{BW}}} - B_i$.

The QDB problem can be solved by any algorithm for the classic DB since the same regret is used between them. Algorithms for the DB problem specify two arms (i, j) to compare at each round and receive a result of the noisy comparison generated from $\mathrm{Ber}(\mu_{i,j})$, where $\mathrm{Ber}(p)$ is the Bernoulli distribution with success probability p. This comparison can be simulated in the QDB problem as follows: We observe X_i and X_j by pulling both arms and return which $X_i > X_j$ or $X_i < X_j$ occurred with ties broken at random.

However, in the QDB problem, we can directly estimate $\mu_{i,j}$ from the feedback distribution of each arm, which significantly enhances exploration. Considering that $\{\nu_i\}$ are all categorical distributions on $[L]$, we have another representation for $\mu_{i,j}$ given by

$$\mu_{i,j} = \sum_{k=1}^{L} P_k^{(i)} \left(\sum_{l=1}^{k} P_l^{(j)} - \frac{1}{2} P_k^{(j)} \right),$$

where $P_k^{(i)} = \mathbb{P}[X_i = k]$. Let $\mathcal{P}_L$ be the probability simplex $\mathcal{P}_L = \{\boldsymbol{x} \in [0,1]^L | \sum_{i=1}^{L} x_i = 1\}$, and we define function $\mu : \mathcal{P}_L \times \mathcal{P}_L \to [0,1]$ as

$$\mu(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{L} x_k \left(\sum_{l=1}^{k} y_l - \frac{1}{2} y_k \right). \tag{1}$$

Hence, $\mu(\boldsymbol{P}^{(i)}, \boldsymbol{P}^{(j)}) = \mu_{i,j}$ for $\boldsymbol{P}^{(i)} = (P_1^{(i)}, \dots, P_L^{(i)})^\top$.

4 Qualitative Dueling Bandit with the Condorcet Winner

In this section, we propose an algorithm for the QDB problem with the Condorcet winner. The algorithm is called *Thompson Condorcet sampling*, which is based on Thompson sampling (Thompson 1933), an algorithm famous for its good performance in the standard MAB problem and wide applicability to many other problems.

Algorithm 1: Thompson Condorcet sampling

```
Set C^(i) = 0 for all i ∈ [K];
Pull all arms t_0 times, update C^(i);
foreach t = Kt_0, Kt_0 + 1, ..., T do
    For each arm i, sample θ^(i) from
      Dir(C_1^(i) + 1, ..., C_L^(i) + 1);
    if ∃i : μ(θ^(i), θ^(j)) ≥ 1/2 for all j ∈ [K] then
        Pull arm a_t = i, observe reward r_t;
        Set C_{r_t}^(a_t) ← C_{r_t}^(a_t) + 1;
    else
        // If there is no Condorcet
           winner, sample {θ^(i)}_{i=1}^K again.
        Goto Line 4;
```

This algorithm maintains Bayesian posterior distributions of $\boldsymbol{P}^{(i)}$ defined in Section 3. We employ the Dirichlet distribution $\mathrm{Dir}(\alpha_1, \dots, \alpha_L)$ as the prior distribution, the probability density function of which is

$$f(\boldsymbol{\theta}; \alpha_1, \dots, \alpha_L) = \frac{\Gamma\left(\sum_{i=1}^{L} \alpha_i\right)}{\prod_{i=1}^{L} \Gamma(\alpha_i)} \prod_{i=1}^{L} \theta_i^{\alpha_i - 1},$$

where $\Gamma(x)$ is the gamma function.

Having Dirichlet distributions as priors is a convenient choice when observations are sampled from a categorical distribution. Let $\boldsymbol{C}^{(i)}(t) = (C_1^{(i)}(t), \dots, C_L^{(i)}(t))$ be the vector representing the observation until the t-th round, where $C_k^{(i)}(t) \in \{0, 1, \dots\}$ represents the number of times that the feedback $k \in [L]$ is observed when arm $i \in [K]$ is pulled. If we employ the prior distribution as $\mathrm{Dir}(1, \dots, 1)$, then the posterior distribution given observations $\boldsymbol{C}^{(i)}(t)$ is $\mathrm{Dir}(C_1^{(i)}(t) + 1, \dots, C_L^{(i)}(t) + 1)$. For notational simplicity, we sometimes denote $\boldsymbol{C}^{(i)}(t)$ as $\boldsymbol{C}^{(i)}$ when the round t is obvious from the context.

The entire algorithm is shown in Algorithm 1. At each round t, the algorithm samples $\boldsymbol{\theta}^{(i)}$ from the posterior distributions of $\boldsymbol{P}^{(i)}$. Then, the agent tries to pull the Condorcet winner assuming $\boldsymbol{P}^{(i)} = \boldsymbol{\theta}^{(i)}$. If the Condorcet winner does not exist, the algorithm resamples $(\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(L)})$ until it exists.

Let $\boldsymbol{P}^{*(i)}$ be

$$\boldsymbol{P}^{*(i)} = \arg\min_{\boldsymbol{P} \in \mathcal{P}_L} \mathrm{KL}(\boldsymbol{P}^{(i)} \| \boldsymbol{P}) \quad \text{s.t.}\, \mu(\boldsymbol{P}, \boldsymbol{P}^{(a^*_{\mathrm{CW}})}) \geq \frac{1}{2}$$

for Kullback-Leibler (KL) divergence $\mathrm{KL}(\boldsymbol{x} \| \boldsymbol{y}) = \sum_{i=1}^{L} x_i \log \frac{x_i}{y_i}$. Then, the regret of Thompson Condorcet sampling is bounded as follows.

Theorem 1. *If the Condorcet winner exists, then there exists $t_0 > 0$ such that the regret of Thompson Condorcet sampling*

is bounded by

$$\mathbb{E}\left[R_T^{\mathrm{CW}}\right] \leq \sum_{i=1}^{K}(1+\varepsilon)\frac{\Delta_i^{\mathrm{CW}}}{\mathrm{KL}(\boldsymbol{P}^{(i)}\|\boldsymbol{P}^{*(i)})}\log T + O\left((\log\log T)^2\right) + O\left(\frac{1}{\varepsilon^{2L}}\right) \quad (2)$$

for any sufficiently small $\varepsilon > 0$.

The proof is given in the longer version, where the detailed condition on t_0 and the precise form of the bound is also provided. From the precise form of (2) that can be found in the longer version, one can see that this regret bound grows exponentially with the number of arms K. However, this is not the inherent limitation of the Thompson Condorcet sampling but the artifact of pursuing the optimal asymptotic dependence on $O(\log T)$. As we will show in Section 6, this exponential increase in the regret does not occur in pracitice, and the algorithm works well for relatively large K.

The regret bound has a similar form to the information theoretic lower bound in the MAB problems for multi-parameter models (Burnetas and Katehakis 1996). Note that considering distributions $\boldsymbol{P}^{*(i)}$ is essential in these cases, whereas they are replaced with the distribution of the optimal arm in the regret bound of Thompson sampling in the MAB problem with the Bernoulli model given by Agrawal and Goyal (2013). For example, when $\boldsymbol{P}^{(a^*_{\mathrm{CW}})} = (\varepsilon, 1-2\varepsilon, \varepsilon)^\top$ and $\boldsymbol{P}^{(i)} = (0.5, 0.1, 0.4)^\top$, we have $\mathrm{KL}(\boldsymbol{P}^{(i)}\|\boldsymbol{P}^{*(i)})/\mathrm{KL}(\boldsymbol{P}^{(i)}\|\boldsymbol{P}^{(a^*_{\mathrm{CW}})}) \to 0$ as $\varepsilon \to 0$.

Theorem 1 suggests the possibility of Thompson Condorcet sampling performing drastically better than the case when we apply classic DB algorithms for the QDB problem in the way discussed in Section 3. The regret lower bound of such direct applications immediately follows from the lower bound for the classic DB problem given by Komiyama et al. (2015).

Proposition 1 (Adapted from Komiyama et al., 2015)**.** *When we apply any consistent algorithms for the DB problem to the QDB problem, we have*

$$\liminf_{T\to\infty}\frac{\mathbb{E}\left[R_T^{\mathrm{CW}}\right]}{\log T} \geq \sum_{i\neq a^*_{\mathrm{CW}}}\min_{j:\mu_{i,j}<\frac{1}{2}}\frac{\Delta_i^{\mathrm{CW}}+\Delta_j^{\mathrm{CW}}}{d(\mu_{i,j},\frac{1}{2})}, \quad (3)$$

where $d(x,y) = x\log\frac{x}{y} + (1-x)\log\frac{1-x}{1-y}$.

From the upper bound given in Theorem 1, we have

$$\lim_{T\to\infty}\frac{\mathbb{E}\left[R_T^{\mathrm{CW}}\right]}{\log T} \leq (1+\varepsilon)\sum_{i\neq a^*_{\mathrm{CW}}}\frac{\Delta_i^{\mathrm{CW}}}{\mathrm{KL}(\boldsymbol{P}^{(i)}\|\boldsymbol{P}^{*(i)})},$$

which can be arbitrarily smaller than (3) as stated in the next lemma.

Lemma 1. *Assume that* $a^*_{\mathrm{CW}} \neq 1$. *For any fixed* $0 < \varepsilon < 1/(4-4\log 2)$*, there exist* $\boldsymbol{P}^{(a^*_{\mathrm{CW}})}, \boldsymbol{P}^{(1)} \in \mathcal{P}_2$ *such that*

$$\frac{d(\mu(\boldsymbol{P}^{(a^*_{\mathrm{CW}})}, \boldsymbol{P}^{(1)}), 1/2)}{\mathrm{KL}(\boldsymbol{P}^{(1)}\|\boldsymbol{P}^{*(1)})} \leq \varepsilon. \quad (4)$$

Algorithm 2: Thompson Borda sampling

```
Set C^(i) = 0 for all i ∈ [K];
Pull all arms t_0 times, update C^(i);
foreach t = 1, ..., do
    For each arm i, sample θ^(i) from
      Dir(C_1^(i) + 1, ..., C_L^(i) + 1);
    B_i ← 1/(K−1) Σ_{j≠i} μ(θ^(i), θ^(j));
    Pull arm a_t = arg max_{i∈[K]} B_i;
    Observe r_t and set C_{r_t}^(a_t) ← C_{r_t}^(a_t) + 1;
```

The proof can be found in the longer version. From Lemma 1, we can say that there exists a case where Thompson Condorcet sampling can perform arbitrarily better than the direct application of any algorithms in the DB. This implies that the algorithm successfully incorporates the qualitative information to reduce the regret in the DB.

5 Qualitative Dueling Banidt with the Borda Winner

In this section, we study two algorithms for the QDB problem with the Borda winner, the one based on the Thompson sampling called *Thompson Borda sampling* and the other based on the UCB algorithm (Auer 2003) called *Borda-UCB*. In spite of the success of Thompson Condorcet sampling, our theoretical analysis reveals that Thompson Borda sampling can have polynomial regret in some setting. On the other hand, Borda-UCB achieves logarithmic regret, which matches the regret lower bound of the classic DB problems.

Thompson Borda sampling given in Algorithm 2 is similar to Thompson Condorcet sampling. The only difference is that Thompson Borda sampling pulls the Borda winner in samples $(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(L)})$. Since there always exists the Borda winner for any samples $(\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(L)})$, thus we do not need resampling. Although it is works surprisingly well empirically as we will see in Section 6, we prove that it suffers from polynomial regret in the worst case.

Theorem 2. *Assume that there are* $K = 3$ *arms such that arm* 1 *is the Borda winner. Then, there exists* $\boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \boldsymbol{P}^{(3)} \in \mathcal{P}_L$ *such that under Thompson Borda sampling with* $\boldsymbol{\theta}^{(1)} = \boldsymbol{P}^{(1)}$, $\boldsymbol{\theta}^{(2)} = \boldsymbol{P}^{(2)}$, *and* $\boldsymbol{\theta}^{(3)} \sim \mathrm{Dir}(C_1^{(3)}+1, \ldots, C_L^{(3)}+1)$, *the statement*

$$\liminf_{T\to\infty}\frac{\mathbb{E}\left[R_T^{\mathrm{BW}}\right]}{T^\eta} = \xi$$

holds for some constants $\xi, \eta > 0$.

The proof can be found in the longer version. The situation considered in Theorem 2 may be somewhat unrealistic since we assume that $\boldsymbol{P}^{(1)}$ and $\boldsymbol{P}^{(2)}$ are known beforehand. However, we will show by an experiment that Thompson Borda sampling actually suffers from the polynomial regret without such an assumption in Section 6.

Another proposed algorithm, Borda-UCB, is based on the UCB algorithm (Auer 2003), which is shown in Algorithm 3. As in the original UCB algorithm, we consider the

Algorithm 3: Borda-UCB

Set $\boldsymbol{C}^{(i)} = \mathbf{0}$ for all $i \in [K]$ and $N_i = 0$;
Pull all arms τ times and get initial estimations;
while $t \leq T$ **do**
  $\hat{\boldsymbol{P}}^{(i)} \leftarrow \boldsymbol{C}^{(i)}/N_i$ for each arm $i \in [K]$;
  $\hat{B}_i \leftarrow \frac{1}{K-1}\sum_{k\in[K]\setminus\{i\}} \mu(\hat{\boldsymbol{P}}^{(i)}, \hat{\boldsymbol{P}}^{(k)})$;
  $\gamma_i \leftarrow \sqrt{\frac{\alpha \log t}{N_i}}$;
  $\beta_i \leftarrow \gamma_i + \frac{1}{K-1}\sum_{k\in[K]\setminus\{i\}} \gamma_k$;
  $i_{\mathrm{UCB}} \leftarrow \arg\max_{i\in[K]} B_i + \beta_i$;
  $i_{\mathrm{Count}} \leftarrow \{i \in [K] | N_i = \max_{j\in[K]} N_j\}$;
  if $i_{\mathrm{UCB}} \in i_{\mathrm{Count}}$ **then**
    Pull arm $a_t = i_{\mathrm{UCB}}$, observe reward r_t;
    $N_i \leftarrow N_i + 1$, $C_{a_t}^{(r_t)} \leftarrow C_{a_t}^{(r_t)} + 1$;
  else
    Pull all arms in $[K]\backslash i_{\mathrm{Count}}$;
    Update N_i and $C_k^{(i)}$;

upper confidence bound $\hat{B}_i + \beta_i$ for each arm $i \in [K]$, where $\hat{B}_i$ is an estimated Borda score, and β_i is the width of the confidence interval controlled by a positive parameter α. Let i_{UCB} be the arm with the largest upper confidence bound. While the original UCB algorithm always pulls the arm with the largest upper confidence bound, Borda-UCB pulls all arms that do not belong to i_{Count}, the set of arms that were pulled the most, if i_{UCB} does not belong to i_{Count}. This exploration strategy reflects the fact that we have to estimate all feedback distributions accurately in order to have the precise estimation of the Borda score.

The regret of Borda-UCB is bounded as follows.

Theorem 3. *Assume that α is set as*

$$\alpha = \max\left(2, \frac{3(1+3\varepsilon')^2}{2(1-\varepsilon')^2}\left(\frac{K-1}{K-2}\right)^2\right)$$

for arbitrarily taken $\varepsilon' > 0$. Then, for any $\varepsilon > 0$, the regret of Borda-UCB is bounded as

$$\mathbb{E}\left[R_T^{\mathrm{BW}}\right] \leq \Delta_{\mathrm{all}}^{\mathrm{BW}}\left(\frac{4\alpha}{(\Delta_{\mathrm{min}}^{\mathrm{BW}} - 2\varepsilon)^2}\log T + C_\varepsilon + C_{\varepsilon'}\right)$$

for some constants $C_\varepsilon = O\left(\frac{1}{\varepsilon^2}\right), C_{\varepsilon'} = O\left(\frac{1}{(\varepsilon')^2}\right)$, where $\Delta_{\mathrm{all}}^{\mathrm{BW}} = \sum_{i\neq a^*_{\mathrm{BW}}} \Delta_i^{\mathrm{BW}}$ and $\Delta_{\mathrm{min}}^{\mathrm{BW}} = \min_{i\neq a^*_{\mathrm{BW}}} \Delta_i^{\mathrm{BW}}$.

The proof is presented in the longer version, where the explicit forms of C_ε and C'_ε are also provided. The regret bound in Theorem 3 is simplified to $O(K\Delta^{-2}\log T)$ when $\Delta_i^{\mathrm{BW}} = \Delta$ for all $i \neq i^*$, while the regret of the original UCB algorithm is $O(K\Delta^{-1}\log T)$ (Auer 2003), which is smaller by $O(1/\Delta)$. However, this difference is inevitable, as proved in the following theorem.

Theorem 4. *Consider two instances of the QDB problem with $K = 3$, in which the feedback distributions of the arms are represented as $\Gamma = (\boldsymbol{P}_\Gamma^{(1)}, \boldsymbol{P}_\Gamma^{(2)}, \boldsymbol{P}_\Gamma^{(3)})$ and $\Theta = (\boldsymbol{P}_\Gamma^{(1)}, \boldsymbol{P}_\Gamma^{(2)}, \boldsymbol{P}_\Gamma^{(3)})$. Let R_T^Γ and R_T^Θ be the regret in each instance. Then, there exists a pair of instances (Γ, Θ) that all algorithms which achieve*

$$\mathbb{E}\left[R_T^\Gamma\right] \leq o(T^a)$$

for all constant $a > 0$ satisfy

$$\liminf_{T\to\infty} \frac{\mathbb{E}\left[R_T^\Theta\right]}{\log T} = \Omega\left(\frac{1}{(\Delta_{\mathrm{min}}^{\mathrm{BW}})^2}\right),$$

where $\Delta_{\mathrm{min}}^{\mathrm{BW}} = \min_{i\neq a^*_{\mathrm{BW}}} \Delta_i^{\mathrm{BW}}$ defined on Θ.

The proof is presented in the longer version. This theorem states that if the algorithm achieves sub-polynomial regret for all instances of the QDB problem with the Borda winner, there exists a case where it suffers from $\Omega((\Delta_{\mathrm{min}}^{\mathrm{BW}})^{-2}\log T)$ regret. Therefore, we can conclude that the difference in the regret upper-bound between the original UCB and Borda-UCB comes from the characteristic of the QDB problem.

The upper bound in Theorem 3 matches the regret lower bound in the classic DB problem, which is considered in the context of the δ-PAC DB problem (Jamieson et al. 2015). The algorithm is called δ-PAC if it finds the Borda winner with failure probability less than δ. We have the following bound of the minimum number of samples required in such δ-PAC algorithms.

Proposition 2 (Theorem 1; Jamieson et al., 2015)**.** *Let τ be the total number of pulls. If $K \geq 4$ and $3/8 \leq \mu_{i,j} \leq 5/8$ for all $i, j \in [K]$, then any δ-PAC DB algorithm with $\delta \leq 0.15$ has*

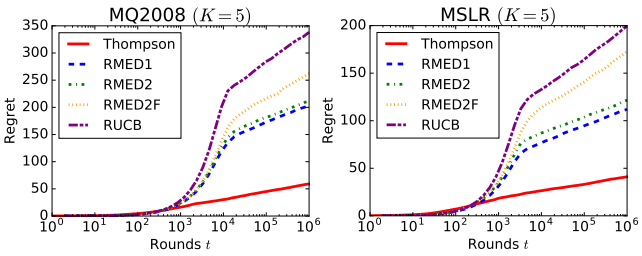
$$\mathbb{E}[\tau] \geq \frac{1}{90}\log\frac{1}{2\delta}\sum_{i\neq a^*_{\mathrm{BW}}}\frac{1}{(\Delta_i^{\mathrm{BW}})^2}.$$

Existing algorithms for the Borda winner (Busa-Fekete et al. 2013; Jamieson et al. 2015) use a δ-PAC DB algorithm as a sub-routine. They first run such an algorithm with $\delta = 1/T$ and then pulls the estimated Borda winner in the remaining rounds. Therefore, the regret of such algorithms is at least $\Omega((\log T)\sum_{i\neq a^*_{\mathrm{BW}}}(\Delta_i^{\mathrm{BW}})^{-2})$ from Proposition 2, and hence the regret upper bound of Borda-UCB is no worse than this lower bound.

Although we were not able to prove that the regret of Borda-UCB is smaller than the direct application of classic DB algorithms, Borda-UCB performs better than them empirically as we will see in Section 6. Furthermore, Borda-UCB has an another advantage that it does not require to specify T. Since existing algorithms run a $(1/T)$-PAC algorithm, it requires the number of rounds T to be known beforehand. However, it is often difficult to guess T beforehand, and thus our algorithms are more useful in practice.

6 Experiments

We test the empirical performance of the proposed algorithms through experiments based on both synthetic setting and real-world data. We first conduct the experiments based on the real-world web search dataset that is also used in the previous work. In the experiments, our methods significantly outperform the direct application of the existing algorithms for the classic DB. Then, we show the results of the experiments in a synthetic setting that Thompson Borda sampling has polynomial regret.



(a) (b)

Figure 1: The regret of Thompson Condorcet sampling and other classic DB algorithms.

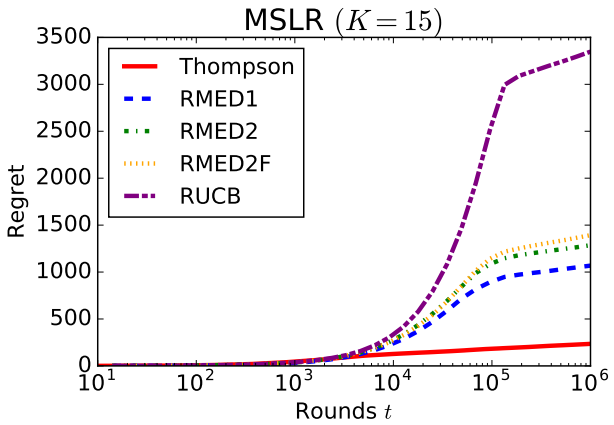
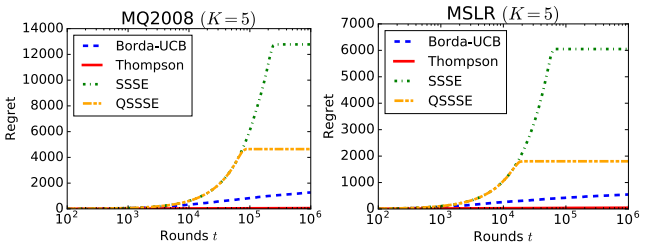


Figure 2: The regret of Thompson Condorcet sampling and other DB algorithms when there are a relatively large number of arms ($K = 15$).



(a) (b)

Figure 3: The regret of Thompson Borda sampling and Borda-UCB with other classic DB algorithms.

Experiments on a Real-World Dataset

We apply proposed methods to the problem of ranker evaluation from the field of information retrieval, which is used for evaluating the algorithms for the classic DB problem in Jamieson et al. (2015). The task is to identify the best ranker, which takes a user's search query as input and ranks the documents according to their relevance to that query.

We used two web search datasets. The first is the MSLR-WEB10K dataset (Qin et al. 2010), which consists of 10,000 search queries over the documents from search results. The data also contains the values of 136 features and a corresponding user-labeled relevance factor on a scale of one to five with respect to each query-document pair. The other is the MQ2008 dataset (Qin and Liu 2013) that contains 46 features and a relevance factor labelled from one to three for each query-document pair. As in Jamieson et al. (2015), we only consider rankers that use one feature to rank documents. Therefore, the aim of the task is to determine which feature is the most capable of predicting the relevance of query-document pairs.

Although Jamieson et al. (2015) set up the classic DB problem from these datasets, we can naturally formulate the QDB problem as well since we have access to the relevance factors. The qualitative feedback is generated in the following way. At each round, the algorithm selects one ranker, and it ranks the documents for a randomly chosen query. The relevance factor for the top-ranked document is revealed to the algorithm as the qualitative feedback. Therefore, we have $L = 5$ in the MSLR-WEB10K dataset and $L = 3$ in the MQ2008 dataset. We compare the regrets of the proposed algorithms to the direct application of the classic DB algorithms, which corresponds to the experiments conducted in Jamieson et al. (2015). We repeat 100 runs for each instance and the mean of the regret is reported.

Experiments for Condorcet Winner We first show the experimental result of the QDB problem with the Condorcet winner. We compare Thompson Condorcet sampling with RUCB (Zoghi et al. 2014), RMED1, RMED2, RMED2F (Komiyama et al. 2015), which are all promising algorithms proposed for the classic DB problem with the Condorcet winner. We set $t_0 = 10$, and the Figure 1 is the experimental result when the number of rankers is $K = 5$.

Figure 1 shows the superiority of Thompson Condorcet sampling. Furthermore, we can observe all existing algorithms incur the large regrets in early rounds while Thompson Condorcet sampling does not. This is because most algorithms for the DB problem construct a set of candidates for the Condorcet winner and explores it in the first part of the rounds, but Thompson Condorcet sampling conducts exploration and exploitation at the same time and does not require such a set. In this sense, Thompson Condorcet sampling performs more stably than the existing methods.

To see the dependency of the performance of Thompson Condorcet sampling on the number of arms, we tried the setting in which we have a relatively large number of arms. The result is shown in Figure 2, in which Thompson Condorcet sampling still performs the best among the other classic DB algorithms even though the regret upper-bound proved in Theorem 1 grows exponentially with K. This result supports the argument that exponential dependency on K is just an artifact of pursuing the best regret bound in the asymptotic case and Thompson Condorcet sampling empirically performs much better than the theoretical analysis.

Experiments for Borda Winner For the Borda setting, we compare our proposed methods, Thompson Borda Sampling and Borda-UCB, with existing classic DB algorithm SSSE (Busa-Fekete et al. 2013). Furthermore, we also con-

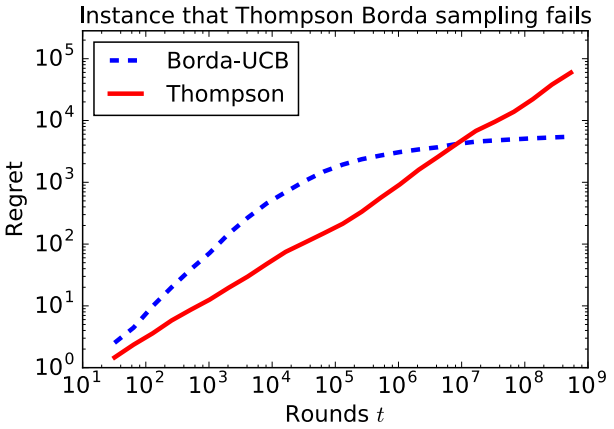


Figure 4: The regret of proposed algorithms in the instance that Thompson Borda sampling suffers the polynomial regret.

duct a comparison with an extension of SSSE, which we call QSEEE, proposed in Busa-Fekete et al. (2013) to utilize the qualitative feedback explicitly.

The result is shown in Figure 3, which shows the superiority of the proposed methods. As in the Condorcet case, SSSE and QSSSE suffer from a large regret in the early stage, while regret always increases logarithmically in the proposed algorithms. This is because existing methods first only explore, while proposing methods always balance exploration and exploitation. Although existing methods achieve zero-regret after the exploration, this does not mean that they perform better than Borda-UCB in $T \to \infty$ since they require longer exploration phase.

Surprisingly, Thompson Borda sampling works quite well in this setting, even though Theorem 2 states that it has the polynomial regret in the worst case. We suspect it is rare to encounter such a worst case in practice, but the condition for sub-polynomial regret is unknown and left to future work.

Experiments on a Synthetic Setting

Theorem 2 proves that Thompson Borda sampling can incur polynomial regret for some instances, which we confirm through experiments in the following. We set up the instance with $K = 3$ and $L = 4$, in which each feedback distribution is represented as $\boldsymbol{P}^{(1)} = (0.0, 0.0, 1.0, 0.0)^\top$, $\boldsymbol{P}^{(2)} = (0.0, 0.5, 0.0, 0.5)^\top$, and $\boldsymbol{P}^{(3)} = (0.2, 0.4, 0.3, 0.1)^\top$. Here, the Borda winner is $a^*_{\mathrm{BW}} = 1$. We repeat running Thompson Borda sampling and Borda-UCB in this instance for 10 times, and the mean of regret is shown in Figure 4.

From Figure 4, we can clearly see that Thompson Borda sampling suffers from polynomial regret, while Borda-UCB still has sub-polynomial regret. However, it takes many rounds for Borda-UCB to have less regret than Thompson Borda sampling. This is because Thompson Borda sampling explores less than necessary. In early rounds, UCB-Borda pulls arm 3 many times, which is necessary for knowing the Borda winner but incurs large regret. On the other hand, Thompson Borda sampling exploits arms 1 and 2 more, which leads its superior performance in early rounds.

7 Conclusions

In this paper, we formulated and studied a novel type of the dueling bandit, called a qualitative dueling bandit. In this problem, an agent receives qualitative feedback at each round and aims to minimize the same regret as the classic DB when the duel is carried out based on that feedback.

We considered two notions of winners, the Condorcet winner and the Borda winner. For the Condorcet winner, we proposed an algorithm, called Thompson Condorcet sampling, and we showed that the regret can be arbitrarily smaller than the direct application of the algorithms in classic DB. Thompson Condorcet sampling also exhibited the superior performance in the experiments based on the real-word web search datasets.

For the Borda winner, we studied two algorithms, Thompson Borda sampling and UCB-Borda. Although the theoretical analysis reveals that Thompson Borda sampling can have polynomial regret in some instances, the experiments showed that it performs surprisingly well empirically, especially when the number of rounds is not very large. On the other hand, we prove the logarithmic regret upper bound for UCB-Borda, which is no worse than the regret lower bound in the classic DB.

As future work, it is important to derive general algorithms that can handle various notions of winners as in Ramamohan, Rajkumar, and Agarwal (2016). Another promising direction is to improve the algorithms for the Borda winner and achieve regret significantly smaller than the classic DB as Thompson Condorcet sampling does in the Condorcet winner case.

Acknowledgements

LX utilized the facility provided by Masason Foundation. JH acknowledges support by KAKENHI 18K17998, and MS acknowledges support by KAKENHI 17H00757.

References

Agrawal, S., and Goyal, N. 2013. Further optimal regret bounds for Thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, 99–107.

Auer, P. 2003. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3:397–422.

Burnetas, A. N., and Katehakis, M. N. 1996. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics* 17(2):122–142.

Busa-Fekete, R.; Szörényi, B.; Weng, P.; Cheng, W.; and Hüllermeier, E. 2013. Top-k selection based on adaptive sampling of noisy preferences. In *Proceedings of the 30th International Conference on Machine Learning*, 1094–1102.

Charon, I., and Hudry, O. 2010. An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals of Operations Research* 175(1):107–158.

Hofmann, K.; Whiteson, S.; and de Rijke, M. 2011. A probabilistic method for inferring preferences from clicks. In

Proceedings of the 20th International Conference on Information and Knowledge Management, 249–258.

Jamieson, K.; Katariya, S.; Deshpande, A.; and Nowak, R. 2015. Sparse dueling bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 416–424.

Komiyama, J.; Honda, J.; Kashima, H.; and Nakagawa, H. 2015. Regret lower bound and optimal algorithm in dueling bandit problem. In *Proceedings of The 28th Conference on Learning Theory*, 1141–1154.

Qin, T., and Liu, T. 2013. Introducing LETOR 4.0 datasets. *ArXiv* abs/1306.2597.

Qin, T.; Liu, T.-Y.; Xu, J.; and Li, H. 2010. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval* 13(4):346–374.

Ramamohan, S. Y.; Rajkumar, A.; and Agarwal, S. 2016. Dueling bandits: Beyond Condorcet winners to general tournament solutions. In *Advances in Neural Information Processing Systems 29*, 1253–1261.

Rothschild, M. 1974. A two-armed bandit theory of market pricing. *Journal of Economic Theory* 9:185 – 202.

Szorenyi, B.; Busa-Fekete, R.; Weng, P.; and Hüllermeier, E. 2015. Qualitative multi-armed bandits: A quantile-based approach. In *Proceedings of the 32nd International Conference on Machine Learning*, 1660–1668.

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

Urvoy, T.; Clerot, F.; Féraud, R.; and Naamane, S. 2013. Generic exploration and K-armed voting bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 1191–1199.

Villar, S. S.; Bowden, J.; and Wason, J. 2015. Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statistical Science* 30:199–215.

Wu, H., and Liu, X. 2016. Double Thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems 30*, 649–657.

Yue, Y.; Broder, J.; Kleinberg, R.; and Joachims, T. 2012. The k-armed dueling bandits problem. *Journal of Computer and System Sciences* 78(5):1538–1556.

Zhou, Y.; Chen, X.; and Li, J. 2014. Optimal PAC multiple arm identification with applications to crowdsourcing. In *Proceedings of the 31st International Conference on Machine Learning*, 217–225.

Zoghi, M.; Whiteson, S.; Munos, R.; and de Rijke, M. 2014. Relative upper confidence bound for the K-armed dueling bandit problem. In *Proceedings of the 31st International Conference on Machine Learning*, 10–18.

Zoghi, M.; Karnin, Z.; Whiteson, S.; and de Rijke, M. 2015. Copeland dueling bandits. In *Advances in Neural Information Processing Systems 28*, 307–315.