IE617: Course Project

Duelling Bandits

Preliminary Project Report

Satyankar Chandra (22B0967) Anilesh Bansal (22B0928)
Siddharth Verma (22B2153) Harsh Jitendrakumar (23N0452)

Spring 2025

Contents

Guide: Prof. Manjesh Kumar Hanawal
Industrial Engineering and Operations Research, IIT Bombay

1 Introduction

The multi-armed bandit (MAB) problem is a classic framework for sequential decision-making under uncertainty. An agent must repeatedly choose one action (or "arm") from a set of available actions, with the goal of maximizing cumulative reward (or minimizing cumulative regret) over time. In the standard stochastic MAB setting, pulling an arm yields a reward drawn from a stationary but initially unknown probability distribution associated with that arm. The agent must balance between exploration and exploitation and learn the optimal action through repeated trials.

In this project, we explore a variant of the MAB problem known as the **Dueling Bandits** problem. This setting is particularly relevant in scenarios where obtaining absolute rewards is difficult or impossible, but relative feedback through pairwise comparisons is available. Here, the agent selects a pair of arms to "duel" in each round, and the outcome is a noisy signal indicating which arm is preferred. The goal remains to minimize regret, but the challenge lies in efficiently identifying the best arm using only these pairwise comparisons.

1.1 Problem Statement

While the standard MAB framework assumes that the agent observes a quantifiable reward after choosing an arm, many real-world scenarios deviate from this model. Consider applications like:

- **Information Retrieval:** Evaluating the absolute quality of a search engine's ranking for a query is difficult. However, users might implicitly indicate preference between two rankings presented side-by-side (e.g., via clicks on an interleaved result list [1]).
- **Sensory Testing:** Quantifying the "goodness of taste" of a food product on an absolute scale is subjective and unreliable. However, asking testers which of two samples tastes better is a common and relatively reliable practice.
- **Recommendation Systems:** Users may struggle to give an absolute rating to an item, but can often state a preference between two suggested items.

In these situations, obtaining absolute rewards is impractical or impossible, but obtaining *relative feedback* through pairwise comparisons is feasible. This motivates the study of **Dueling Bandits**.

The **K-armed Dueling Bandits Problem**, as formulated by **Yue et al.** [2], addresses this setting.

1.2 Literature Explored

This preliminary report focuses primarily on the foundational work by Yue et al. [2], which formally introduced the K-armed Dueling Bandits problem in a regret-minimization framework and proposed algorithms with theoretical guarantees. This paper establishes the core problem setup, regret definitions, and key algorithmic ideas (Interleaved Filters) that form the basis of our project.

Some other relevant works in the literature include:

(i) [CBLS06] Nicol‘ o Cesa-Bianchi, G´ abor Lugosi, and Gilles Stoltz. Regret minimization under partial monitoring. Mathematics of Operations Research, 31(3):562–580, 2006.
- It defines a partial monitoring problem, a class of regret-minimization problems in which an algorithm (the “forecaster”) chooses actions and then observes feedback signals that depend on the actions chosen by the forecaster and by an unseen opponent (the “environment”).

(ii) [EDMM06] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research (JMLR), 7:10791105, 2006.

(iii) [FRPU94] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. SIAM Journal on Computing, 23(5), 1994.

(iv) [YJ09] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In International Conference on Machine Learning (ICML), 2009.

2 The Duelling Bandit Problem

This section formalizes the K-armed Dueling Bandits problem as presented in [2].

2.1 Problem Setup

We are given a set of K bandits (arms or strategies), denoted by $\mathcal{B} = \{b_1, b_2, \ldots, b_K\}$. In each iteration $t = 1, 2, \ldots, T$, where T is the time horizon, the learning algorithm selects a pair of bandits $(b_i^{(t)}, b_j^{(t)})$ to compare.

The outcome of the comparison is a random variable indicating which bandit "won" the duel. It is assumed that the probability of bandit b_i winning against bandit b_j is stationary over time and is denoted by:

$$\mathbb{P}(b_i > b_j)$$

We define the preference margin $\epsilon(b_i, b_j) \in (-\frac{1}{2}, \frac{1}{2})$ as:

$$\epsilon(b_i, b_j) = \mathbb{P}(b_i > b_j) - \frac{1}{2}$$

Note that $\epsilon(b_i, b_j) = -\epsilon(b_j, b_i)$ and $\epsilon(b_i, b_i) = 0$. The value $\epsilon(b_i, b_j)$ captures the degree of distinguishability between b_i and b_j. A positive $\epsilon(b_i, b_j)$ implies b_i is preferred over b_j.

It is assumed that there exists a total ordering $>$ over the bandits $\mathcal{B}$ such that $b_i > b_j$ implies $\epsilon(b_i, b_j) > 0$. The single best bandit according to this ordering is denoted by b^*. The goal of the algorithm is to identify strategies $(b_i^{(t)}, b_j^{(t)})$ over time such that the cumulative regret is minimized.

With the guarantee of a total ordering, we can assume without loss of generality that the bandits are indexed such that $b_1 \geq b_2 \geq \cdots \geq b_K$. Obviously, this ordering is not known to the algorithm, but it is a useful assumption for theoretical analysis.

2.2 Definition of Regret

Since absolute rewards are not available, standard regret definitions are not applicable. Yue et al. [2] propose two natural notions of regret based on the preference margins relative to the best bandit b^*:

1. **Strong Regret (R_T^{strong}):** This measures the expected preference gap between the best arm b^* and *both* arms chosen for comparison at each step.

$$R_T^{strong} = \sum_{t=1}^{T} \left(\epsilon(b^*, b_i^{(t)}) + \epsilon(b^*, b_j^{(t)}) \right) \tag{1}$$

   This can be interpreted as the expected total number of times a user (whose preferences align with the $\mathbb{P}(b_i > b_j)$ probabilities) would have preferred b^* over a randomly chosen member of the pair $(b_i^{(t)}, b_j^{(t)})$.

2. **Weak Regret (R_T^{weak}):** This measures the expected preference gap between the best arm b^* and the *better* of the two arms chosen for comparison at each step.

$$R_T^{weak} = \sum_{t=1}^{T} \min \left\{ \epsilon(b^*, b_i^{(t)}), \epsilon(b^*, b_j^{(t)}) \right\} \tag{2}$$

   This represents the expected total number of times a user would have preferred b^* over the winner (or empirically superior arm) of the duel $(b_i^{(t)}, b_j^{(t)})$.

The algorithms presented aim to minimize these regret measures. Often, bounds derived for strong regret also apply (up to constants) to weak regret under certain assumptions.

2.3 Assumptions

To facilitate theoretical analysis, the paper imposes structure on the pairwise comparison probabilities $\mathbb{P}(b_i > b_j)$ through two key assumptions:

1. **Strong Stochastic Transitivity (SST):** For any triplet of bandits $b_i > b_j > b_k$, the preference margin between the best and worst is at least as large as the margins in the intermediate comparisons:

$$\epsilon(b_i, b_k) \geq \max\{\epsilon(b_i, b_j), \epsilon(b_j, b_k)\} \tag{3}$$

   This provides a basic monotonicity constraint – the preference for b_i over b_k cannot be weaker than its preference over an intermediate bandit b_j, or the preference of b_j over b_k.

2. **Stochastic Triangle Inequality (STI):** For any triplet of bandits $b_i > b_j > b_k$, the preference margin between the best and worst is no more than the sum of the intermediate margins:

$$\epsilon(b_i, b_k) \leq \epsilon(b_i, b_j) + \epsilon(b_j, b_k) \tag{4}$$

   This assumption captures a notion of diminishing returns. It implies that as a bandit becomes significantly superior to another, the marginal increase in winning probability slows down. It prevents scenarios where a small gap between b_i, b_j and b_j, b_k could lead to an arbitrarily large gap between b_i, b_k.

These assumptions are not overly restrictive and are satisfied by many common models of pairwise comparisons. They ensure that the preference margins behave in a consistent and predictable manner, allowing for effective learning and exploration strategies.

Indeed, these assumptions are satisfied by common preference models, including:

- **Logistic/Bradley-Terry Model:** Each bandit b_i has an associated positive strength μ_i, and $\mathbb{P}(b_i > b_j) = \frac{\mu_i}{\mu_i + \mu_j}$.
- **Thurstone (Gaussian) Model:** Each bandit b_i has an associated random utility $X_i \sim \mathcal{N}(\mu_i, \sigma^2)$. Comparisons are made based on these utilities, $\mathbb{P}(b_i > b_j) = \mathbb{P}(X_i > X_j) = \mathbb{P}(X_i - X_j > 0)$. If X_i are independent Gaussians with equal variance, $X_i - X_j \sim \mathcal{N}(\mu_i - \mu_j, 2\sigma^2)$, and the assumptions hold.

In section 5.1, we will discuss some concrete use cases where these assumptions are satisfied.

3 Algorithms

The solution proposed in [2] follows an "explore then exploit" strategy. An exploration algorithm identifies the best bandit (with a high probability), and then the exploit phase repeatedly uses that bandit. The core contribution lies in the design of the two exploration algorithms - **Interleaved Filter 1 (IF1)** and **Interleaved Filter 2 (IF2)**.

3.1 General Framework

The high-level approach is outlined in Algorithm 1 of the paper.

Algorithm 1 Explore Then Exploit Solution

1: Input: $T, \mathcal{B} = \{b_1, \ldots, b_K\}, EXPLORE$
2: $(\hat{b}, \hat{T}) \leftarrow EXPLORE(T, \mathcal{B})$
3: **for** $t = \hat{T} + 1, \ldots, T$ **do**
4: compare $\hat{b}$ and $\hat{b}$
5: **end for**

Figure 1: General Explore Then Exploit Framework

The algorithm operates as follows:

1. **Input:** Time horizon T, bandit instance $\mathcal{B}$.
2. **Exploration Phase:** Run an exploration algorithm (e.g., IF1 or IF2) on $\mathcal{B}$. This algorithm performs comparisons and aims to identify the best bandit b^*. Let the algorithm return its estimate $\hat{b}$ and the total number of comparisons (time steps) it used, $\hat{T}$.
3. **Exploitation Phase:** If $\hat{T} \leq T$, then for the remaining $t = \hat{T} + 1, \ldots, T$ iterations, repeatedly compare the estimated best bandit $\hat{b}$ with itself (i.e., select the pair $(\hat{b}, \hat{b})$). This incurs zero regret if $\hat{b} = b^*$. If $\hat{T} > T$, the algorithm terminates without any exploitation phase but the regret bounds still apply.

The total regret is dominated by the regret incurred during the exploration phase, plus potentially large regret if the exploration phase makes a mistake and returns $\hat{b} \neq b^*$ (though this happens with low probability). The focus is therefore on designing efficient exploration algorithms with low regret and low probability of error.

Both the IF1 and IF2 algorithms return a single bandit arm $\hat{b}$ as the best bandit. We will see in section 4.2 that the best arm is returned with probability at least $1 - O(1/T)$.

3.2 Interleaved Filter 1 (IF1)

Interleaved Filter 1 (Algorithm 2 in [2]) is the first proposed exploration algorithm. Its key ideas are:

- **Candidate Bandit:** It maintains a current candidate bandit $\hat{b}$, initially chosen randomly.
- **Active Set:** It maintains a set W of challenger bandits, initially $\mathcal{B} \setminus \{\hat{b}\}$.
- **Interleaved Comparisons:** In each "round" associated with the candidate $\hat{b}$, it performs one comparison between $\hat{b}$ and *each* bandit $b \in W$. This round-robin or interleaved approach ensures all active challengers are compared against the current candidate relatively quickly.
- **Empirical Estimates & Confidence Intervals:** For each pair $(\hat{b}, b)$ with $b \in W$, it maintains an empirical estimate $\hat{P}_{\hat{b},b}$ of $\mathbb{P}(\hat{b} > b)$ based on the comparisons performed so far within the current round. It also maintains a confidence interval $\hat{C}_{\delta,\hat{b},b}$ around this estimate, typically based on Hoeffding's inequality, where δ is a small confidence parameter (e.g., $\delta = 1/(TK^2)$).
- **Elimination Rule:** After each comparison involving $b \in W$, IF1 checks if the confidence interval $\hat{C}_{\delta,\hat{b},b}$ lies entirely above $1/2$. If $\hat{P}_{\hat{b},b} > 1/2$ and $1/2 \notin \hat{C}_{\delta,\hat{b},b}$, it means we have high confidence that $\hat{b} > b$. In this case, bandit b is eliminated from the active set W.

Algorithm 2 Interleaved Filter 1 (IF1)

Input: $T, \mathcal{B} = \{b_1, \ldots, b_K\}$
$\delta \leftarrow 1/(TK^2)$
Choose $\hat{b} \in \mathcal{B}$ randomly
$W \leftarrow \{b_1, \ldots, b_K\} \setminus \{\hat{b}\}$
$\forall b \in W$, maintain estimate $\hat{P}_{\hat{b},b}$ of $P(\hat{b} > b)$
$\forall b \in W$, maintain $1-\delta$ confidence interval $\hat{C}_{\hat{b},b}$ of $\hat{P}_{\hat{b},b}$
while $W \neq \emptyset$ **do**
  for $b \in W$ **do**
    compare $\hat{b}$ and b
    update $\hat{P}_{\hat{b},b}$, $\hat{C}_{\hat{b},b}$
  end for
  while $\exists b \in W$ s.t. $\left(\hat{P}_{\hat{b},b} > 1/2 \wedge 1/2 \notin \hat{C}_{\hat{b},b}\right)$ **do**
    $W \leftarrow W \setminus \{b\}$
  end while
  if $\exists b' \in W$ s.t. $\left(\hat{P}_{\hat{b},b'} < 1/2 \wedge 1/2 \notin \hat{C}_{\hat{b},b'}\right)$ **then**
    $\hat{b} \leftarrow b'$, $W \leftarrow W \setminus \{b'\}$ *//new round*
    $\forall b \in W$, reset $\hat{P}_{\hat{b},b}$ and $\hat{C}_{\hat{b},b}$
  end if
end while
$\hat{T} \leftarrow$ Total Comparisons Made
return $(\hat{b}, \hat{T})$

Figure 2: Interleaved Filter 1 (IF1)

Algorithm 3 Interleaved Filter 2 (IF2)

Input: $T, \mathcal{B} = \{b_1, \ldots, b_K\}$
$\delta \leftarrow 1/(TK^2)$
Choose $\hat{b} \in \mathcal{B}$ randomly
$W \leftarrow \{b_1, \ldots, b_K\} \setminus \{\hat{b}\}$
$\forall b \in W$, maintain estimate $\hat{P}_{\hat{b},b}$ of $P(\hat{b} > b)$
$\forall b \in W$, maintain $1-\delta$ confidence interval $\hat{C}_{\hat{b},b}$ of $\hat{P}_{\hat{b},b}$
while $W \neq \emptyset$ **do**
  for $b \in W$ **do**
    compare $\hat{b}$ and b
    update $\hat{P}_{\hat{b},b}$, $\hat{C}_{\hat{b},b}$
  end for
  while $\exists b \in W$ s.t. $\left(\hat{P}_{\hat{b},b} > 1/2 \wedge 1/2 \notin \hat{C}_{\hat{b},b}\right)$ **do**
    $W \leftarrow W \setminus \{b\}$
  end while
  if $\exists b' \in W$ s.t. $\left(\hat{P}_{\hat{b},b'} < 1/2 \wedge 1/2 \notin \hat{C}_{\hat{b},b'}\right)$ **then**
    while $\exists b \in W$ s.t. $\hat{P}_{\hat{b},b} > 1/2$ **do**
      $W \leftarrow W \setminus \{b\}$ *//pruning*
    end while
    $\hat{b} \leftarrow b'$, $W \leftarrow W \setminus \{b'\}$ *//new round*
    $\forall b \in W$, reset $\hat{P}_{\hat{b},b}$ and $\hat{C}_{\hat{b},b}$
  end if
end while
$\hat{T} \leftarrow$ Total Comparisons Made
return $(\hat{b}, \hat{T})$

Figure 3: Interleaved Filter 2 (IF2)

- **Candidate Update Rule:** IF1 also checks if any challenger $b' \in W$ is statistically significantly better than the current candidate $\hat{b}$. If for some b', $\hat{P}_{\hat{b},b'} < 1/2$ and $1/2 \notin \hat{C}_{\delta,\hat{b},b'}$, it means we have high confidence that $b' > \hat{b}$. In this case, the current candidate $\hat{b}$ is discarded, b' becomes the new candidate ($\hat{b} \leftarrow b'$), the active set is reset to $W \setminus \{b'\}$, and all empirical estimates and counts are reset for the new round.
- **Termination:** The algorithm terminates when the active set W becomes empty, meaning the current candidate $\hat{b}$ has been found to be statistically significantly better than all other bandits. This final $\hat{b}$ is returned.

3.3 Interleaved Filter 2 (IF2)

Interleaved Filter 2 (Algorithm 3 in [2]) builds upon IF1 by adding an optimization step called "pruning" to potentially speed up the elimination process.

- **Core Mechanism:** IF2 operates similarly to IF1 regarding maintaining a candidate $\hat{b}$, an active set W, performing interleaved comparisons, and using confidence intervals for elimination and candidate updates.
- **Pruning Step (Lines 16-18 in Alg 3):** The key difference occurs when a challenger b' is found to be statistically significantly better than the current candidate $\hat{b}$ (triggering a candidate update). Before setting $\hat{b} \leftarrow b'$ and resetting counts, IF2 performs an additional check: For all other bandits $b'' \in W \setminus \{b'\}$, it checks if the empirical estimate $\hat{P}_{\hat{b},b''}$ (from the round that just ended) was greater than $1/2$. If $\hat{P}_{\hat{b},b''} > 1/2$, it means b'' was empirically worse than the just-defeated candidate $\hat{b}$. The pruning step immediately removes such bandits b'' from W.
- **Rationale:** The intuition is that if b' is significantly better than $\hat{b}$, and $\hat{b}$ was empirically better than b'', then by transitivity (especially under the assumed properties), b' is likely much better than b''. Pruning these b'' immediately avoids potentially many comparisons needed later to eliminate them statistically against the new, stronger candidate b'. This aims to reduce the total number of comparisons, especially when the initial candidate is far from optimal.

This pruning step allows IF2 to achieve a better theoretical regret bound compared to IF1.

4 Theoretical Analysis and Key Results

The paper [2] provides theoretical guarantees for both the proposed algorithms, focusing on bounding the probability of making mistakes and the resulting cumulative regret.

4.1 Terminology

We first define some terms used in the paper:

- **Mistake:** The algorithm makes a mistake if it draws a false conclusion about the relative ordering of two bandit arms with high confidence (e.g., concluding $b_i > b_j$ when $b_j > b_i$, or eliminating b^*).
- **Match:** A "match" between two bandit arms b_i, b_j refers to the sequence of comparisons performed between them within a single round (i.e., while one of them is the candidate $\hat{b}$). A match ends when one is eliminated or one replaces the other as the candidate.
- **Round:** A "round" consists of all the matches played concurrently while a particular bandit arm $\hat{b}$ is the candidate.
- **Confidence Interval (CI):** The interval $\hat{C}_{\delta,b_i,b_j} = (\hat{P}_{i,j} - c_t, \hat{P}_{i,j} + c_t)$ where $\hat{P}_{i,j}$ is the empirical win rate of b_i over b_j after t comparisons in the match, and $c_t = \sqrt{\log(1/\delta)/t}$ is the confidence radius derived from Hoeffding's inequality. Lemma 1 shows that the true probability $\mathbb{P}(b_i > b_j)$ lies within this interval with probability at least $1-\delta$. The stopping conditions for matches are based on whether $1/2$ is contained within this interval. Lemma 1 also bounds the maximum number of comparisons t needed in a match with high probability, showing it scales as $O(\log(TK)/\epsilon(b_i, b_j)^2)$.

4.2 Mistake Bounds

A crucial part of the analysis is bounding the probability that the exploration algorithm makes a mistake and fails to return the true best bandit b^*. Note that we are directly reporting the results from the lemmas and theorems in the paper, without going into the details of the proofs.

- **IF1 Mistake Bound (Lemma 3):** By setting the confidence parameter $\delta = 1/(TK^2)$, the probability of making a mistake in any single match is at most δ (from Lemma 1). Since there are at most K^2 potential matches in total, a union bound shows that the probability of IF1 making *any* mistake throughout its execution is at most $K^2\delta = K^2/(TK^2) = 1/T$.
- **IF2 Mistake Bound (Lemma 6, Theorem 3):** The analysis for IF2 is slightly more complex due to the pruning step. Theorem 3 shows that the pruning step itself is safe: if b' defeats $\hat{b}$ with $1-\delta$ confidence, then any b'' that was empirically worse than $\hat{b}$ (i.e., $\hat{P}_{\hat{b},b''} > 1/2$) can be concluded to be worse than b' also with $1-\delta$ confidence, leveraging the STI assumption. Lemma 6 combines this with the basic match mistake probability, concluding that the overall probability of IF2 making *any* kind of mistake (either in a standard comparison or via pruning) is also bounded by $O(1/T)$ when $\delta = O(1/(TK^2))$.

Therefore, both algorithms return the correct best bandit b^* with high probability (at least $1 - O(1/T)$).

4.3 Regret Bounds

The main results are the regret bounds achieved by the algorithms during the exploration phase. The expected total regret (as defined in Section 2.2) is bounded by combining the low probability of making a mistake with the regret accumulated when no mistake is made. The paper also provides an information-theoretic lower bound on the regret, showing that the IF2 algorithm is asymptotically optimal.

- **Regret per Match (Lemma 2):** Assuming no mistakes, the number of comparisons in a match between b_i and b_j is $O(\log(T)/\epsilon_{i,j}^2)$ w.h.p. Under STI, the strong regret incurred during such a match is bounded by $O(\log T/\epsilon_{1,\max(i,j)})$, where $\epsilon_{1,k} = \epsilon(b^*, b_k)$.
- **Number of Rounds/Matches (Lemma 5 for IF1, Lemma 7 for IF2):**

  - IF1 requires $O(\log K)$ rounds with high probability (modeled as a random walk on bandits). Since each round involves $O(K)$ matches, the total number of matches is roughly $O(K \log K)$.
  - IF2, due to pruning, requires only $O(K)$ matches *in expectation* (Lemma 7).

- **Total Regret Bounds:**
  - **Theorem 1 (IF1):** Combining the regret per match and the number of matches ($O(K \log K)$), IF1 incurs expected regret (both strong and weak) bounded by:

$$E[R_T^{IF1}] = \mathcal{O}\left(\frac{K \log K}{\epsilon_{1,2}} \log T\right)$$

    where $\epsilon_{1,2} = \epsilon(b^*, b_2)$ is the gap between the best and second-best bandit (which lower bounds other relevant gaps due to SST). This bound holds with high probability.
  - **Theorem 2 (IF2):** Combining the regret per match and the expected number of matches ($O(K)$), IF2 achieves a better expected regret bound:

$$E[R_T^{IF2}] = \mathcal{O}\left(\frac{K}{\epsilon_{1,2}} \log T\right)$$

- **Lower Bound (Theorem 4):** The paper also provides an information-theoretic lower bound, showing that any algorithm for this problem must incur regret of at least:

$$R_T = \Omega\left(\frac{K}{\epsilon} \log T\right)$$

  where ϵ is related to the minimum preference margin. This demonstrates that the expected regret bound achieved by **IF2 is asymptotically optimal** up to constant factors.

5 Applications and Planned Experiments

5.1 Potential Applications

Some particular examples of applications where the Dueling Bandits framework is relevant include - Search Engine Optimization, Recommendation Systems, A/B Testing Variants, User Interface Design, Personalized Medicine (Drug Testing), and Sensory Analysis.

In our project, we would like to focus on a variant of **Drug Discovery** problem by modelling a subpart of it in the Dueling Bandits framework. In this problem, we are provided with some n generative models (which we can think of as bandit arms) and each model stochastically produces a set of drug molecules. The goal is to identify the best generative model (bandit arm) that produces the most effective and least toxic drug molecules. The challenge is that we can only compare two generative models at a time, and the feedback is based on pairwise comparisons of the drug molecules generated by these models. Hence, this is a natural fit for the Dueling Bandits framework.

We have fully detailed the problem in section 7 of the report.

5.2 Planned Experiments

To empirically validate the theoretical findings and compare the proposed algorithms, we plan to conduct simulation experiments.

1. **Algorithms to Implement:** We will implement Interleaved Filter 1 (IF1) and Interleaved Filter 2 (IF2) as described in [2].
2. **Simulation Environment:**
   - We will simulate the pairwise comparisons based on underlying "quality" scores μ_i for each bandit b_i.
   - We will primarily use the **Gaussian (Thurstone) Model**: Assume $\mu_1 > \mu_2 > \cdots > \mu_K$. The probability $\mathbb{P}(b_i > b_j)$ will be calculated as $\Phi\left(\frac{\mu_i - \mu_j}{\sqrt{2}\sigma}\right)$, where Φ is the standard normal CDF and σ^2 is the variance (we can set $\sigma = 1$).
3. **Parameters to Vary:**
   - Number of bandits (K): e.g., $K = 5, 10, 20, 50$.
   - Time horizon (T): e.g., $T = 10^4, 10^5, 10^6$.
   - Preference Margin/Distinguishability: Particularly focusing on the gap $\epsilon_{1,2} = \mathbb{P}(b_1 > b_2) - 1/2$ by adjusting $\mu_1 - \mu_2$.
   - Noise level (implicitly via σ in the Gaussian model, or inherent in the probability calculation).
4. **Performance Metrics:**
   - **Cumulative Regret:** Plot cumulative strong regret R_t^{strong} (or weak regret R_t^{weak}) vs. time t.
   - **Probability of Error:** Measure the frequency (over multiple runs) with which the algorithms fail to identify b^* at the end of the exploration phase.
   - **Number of Comparisons:** Record the total number of comparisons ($\hat{T}$) used by the exploration phase.
5. **Experimental Goals:**
   - Verify that cumulative regret grows roughly logarithmically with T and linearly with K (or $K \log K$ for IF1).
   - Demonstrate the performance difference between IF1 and IF2, expecting IF2 to generally have lower regret and potentially terminate faster, especially when K is large or the initial random candidate is poor.
   - Analyze the effect of the difficulty parameter $\epsilon_{1,2}$ on the required number of comparisons and the slope of the regret curve.

- Confirm that the probability of error decreases as T increases.

These experiments will provide empirical insights into the practical performance of IF1 and IF2 and how it aligns with their theoretical guarantees.

Another interesting aspect to explore is the effect of **early exit conditions** in the IF1 and IF2 algorithms. In the original paper, the algorithms are designed to terminate when the active set W becomes empty. However, in practice, it may be beneficial to allow the algorithms to exit early if they are sufficiently confident about the best bandit. We can implement a threshold-based early exit condition and analyze its impact on the regret and number of comparisons.

Our aim for this part is:

- To understand how the early exit condition affects the regret and number of comparisons.
- To quantify the performance gain (if any) from allowing early exits.
- To see how the confidence sets change at various checkpoints during the execution of the algorithms.

6 Conclusion

The K-armed Dueling Bandits problem provides a valuable framework for online learning in settings where only pairwise comparisons are available. This preliminary report has reviewed the problem setup, regret definitions (strong and weak), and key assumptions (SST, STI) as introduced by Yue et al. [2]. We detailed the Interleaved Filter algorithms (IF1 and IF2), highlighting the interleaved comparison strategy and the pruning optimization in IF2.

The theoretical analysis shows that both algorithms identify the best bandit with high probability $(1-O(1/T))$. IF2 achieves an asymptotically optimal expected regret bound of $O((K/\epsilon_{1,2})\log T)$, improving upon IF1's $O((K\log K/\epsilon_{1,2})\log T)$ bound due to its more efficient pruning mechanism.

Our planned work involves implementing IF1 and IF2 and conducting simulation experiments to empirically evaluate their performance, compare them, and validate the theoretical findings under **different parameter settings** (K, T, preference gaps) and **early-exit conditions**. This will form the basis for further exploration of the dueling bandits literature and potential extensions as part of the course project.

We are also interested in exploring the implications of the dueling bandits framework in real-world applications, particularly in **drug discovery**, where the problem of selecting the best generative model for drug molecules aligns well with the principles of pairwise comparisons.

Acknowledgments

We would like to thank our course instructor, Prof. Manjesh K. Hanawal, for setting up the problem statement and providing us with the opportunity to explore this interesting area of research. We also appreciate the feedback and guidance from the TAs of IE617 who were always available to help us with our queries.

7 Appendix: Drug Discovery Problem

Molecular Representation and Generation (Literature Review [3] [4] [5])

Representing molecules is a foundational aspect of data-driven drug discovery. Common formats include the Simplified Molecular Input Line Entry System (SMILES) and graph-based data like Crystallographic Information Files (CIF). SMILES is widely used due to its simplicity but lacks uniqueness, while graph representations provide richer structural detail. Other popular descriptors include Morgan fingerprints, Extended Connectivity Fingerprints (ECFP), and molecular images.

Generative models have been transformative in drug discovery. Recurrent Neural Networks (RNNs), Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Adversarial Autoencoders (AAEs) have been widely explored. A significant study showed that RNNs could innovate molecular structures with superior efficiency to that of VAEs. This reduced the use of VAEs in following years in the domain of molecular generation. Another study showed that LSTM layers helped in greater valid generation of SMILES strings than GRU layers. Additionally, it also showed that bi-directional GRU layers performed poorly when compared to both uni- and bi-directional LSTM. In work on a generative framework in low-data regimes, it was shown that RNNs could generate valid molecules independent of explicitly provided chemical rules. From further studies, it is evident that uni-directional and bi-directional generative RNNs performed significantly better than the aforementioned models. Moreover, it is to be noted that forward RNN outperformed bidirectional RNN in terms of valid molecule generation and had novelty much similar to bidirectional RNN.

In another study, a recurrent neural network was trained on a large unfocused dataset of molecules and then fine-tuned on a smaller library of molecules with desired properties. The generated dataset had 93% valid and 90% unique SMILES sequences, with none of the sequences similar to the original database. Similarly, another framework based on bidirectional RNN for de novo molecular design of CNS drugs showed promising results, generating 90% new structures.

Some recent work addresses the challenges in accurately predicting molecular properties, particularly highlighting the limitations of traditional machine learning approaches such as models of the Quantitative Structure-Property Relationship (QSPR). Because QSPRs depend on preset descriptors that need domain expertise, if useful descriptors are not used, the results might be less than ideal. On the other hand, Graph Neural Networks (GNNs) provide a representation learning technique that does not require further feature engineering and instead extracts chemical structure straight from the graph. Atoms and bonds are viewed as nodes and edges in this context, and GNNs aggregate data throughout the molecular graph via a message-passing mechanism. This makes it possible for an end-to-end learning process to produce cutting-edge results on problems involving property prediction.

There have been several approaches for molecular property prediction using GNNs. One such approach demonstrates a robust and promising method for molecular property prediction, particularly within the framework of drug discovery. By leveraging an attention-based GNN architecture such as AttentiveFP, this method excels in representing intricate molecular structures. Traditional QSPR methods, though useful, are heavily reliant on domain-specific descriptors, which can limit predictive accuracy if relevant descriptors are overlooked. In contrast, attention-based GNNs facilitate end-to-end learning from raw molecular graph data, reducing dependency on predefined features and offering a more nuanced molecular representation.

Problem Setup

In our work, we utilize a set of LSTM-based generative models as **arms** in a *duel-bandit* framework for the task of molecular generation. Each LSTM model exhibits stochastic behavior, meaning that the output molecules vary with each generation. Therefore, instead of relying on deterministic selection, we compare models in a pairwise fashion based on their performance in molecule generation.

Let $\mathcal{M} = \{M_1, M_2, \ldots, M_{10}\}$ be the initial set of 10 LSTM-based generative models. Each model M_i is trained with a different configuration of hyperparameters and/or on different subsets of the training data.

The generation process from each model is probabilistic. That is, each time a model is queried, it produces a different (but valid) set of SMILES strings. Let $G(M_i)$ denote the set of molecules generated by model M_i during a trial.

Connection to Dueling Bandits

We employ a duel-bandit strategy in which two models, say M_i and M_j, are selected and compared based on their generated molecules. The outcome of each duel is determined by evaluating the following metrics:

- **Validity:** Proportion of syntactically and chemically valid molecules in $G(M_i)$ and $G(M_j)$.
- **Toxicity:** Proportion of toxic molecules, assessed using an external Graph Neural Network (GNN) trained to classify molecules as toxic or non-toxic.

A model is considered the *winner* in a duel if it generates a higher proportion of valid and less toxic molecules. The exact loss function for the bandit problem is something we are still working on.

Relevance to Drug Discovery

In drug discovery, evaluating molecular novelty or toxicity via absolute reward is non-trivial and error-prone. The dueling bandits paradigm provides a natural framework by relying on pairwise preferences (e.g., molecule A is more novel or less toxic than molecule B). This comparative feedback, potentially provided by domain experts or simulations, allows effective learning even in noisy or complex settings.

Integrating dueling bandits with generative models offers a new pathway to discover molecules with optimal trade-offs between properties like novelty, stability, and toxicity—thus contributing significantly to data-driven pharmaceutical innovation.

References

[1] F. Radlinski, M. Kurup, and T. Joachims, "How does clickthrough data reflect retrieval quality?" In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, ACM, 2008, pp. 43–52.

[2] Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, "The K-armed dueling bandits problem," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, ACM, 2009, pp. 1161–1168.

[3] S. F. Sweere, I. Valtchanov, M. Lieu, *et al.*, "Deep learning-based super-resolution and de-noising for xmm-newton images," *Monthly Notices of the Royal Astronomical Society*, vol. 517, no. 3, pp. 4054–4069, 2022.

[4] K. F. Biegasiewicz, J. R. Griffiths, G. P. Savage, J. Tsanaktsidis, and R. Priefer, "Cubane: 50 years later," *Chemical reviews*, vol. 115, no. 14, pp. 6719–6745, 2015.

[5] C. Lipinski and A. Hopkins, "Navigating chemical space for biology and medicine," *Nature*, vol. 432, no. 7019, pp. 855–861, 2004.