Bandits and Probabilistic Methods

Arjun Karanam, Jirayu Burapacheep, Akash Chaurasia, Lilian Chan, William Shabecoff

Agenda

Introduction

Contextual Bandits

Dueling Bandits

Applications

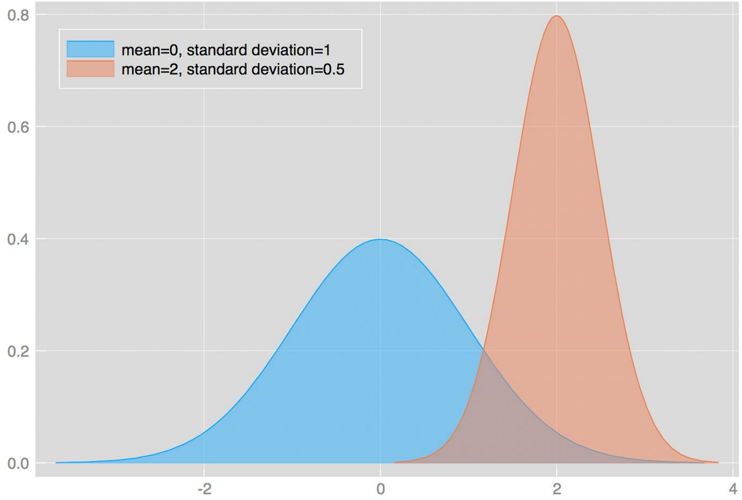
Introduction

What Are Bandits? A Story

One-Armed Bandit

Unknown Reward from Distribution $\mathbf{D_1}$

$\mathbf{D_1 = ?}$



What Are Bandits? A Story

r from Distribution $\mathbf{D}_1$

r from Distribution $\mathbf{D}_2$

r from Distribution $\mathbf{D}_3$

What Are Bandits? Formalization

Notation

You have **K** bandits (or one bandit with **K arms**)

You play this game over **T** rounds

Each arm has a Reward distribution $\mathbf{D_a}$

The Process

For each round $t \in [T]$:

- Your algorithm picks some arm $\boldsymbol{a}$
- Reward $\mathbf{r_t}$ is sampled from $\mathbf{D_a}$
- Your algorithm collects $\mathbf{r_t}$ and observes nothing else

What Are Bandits? A Story

So, let’s say you had 100 tries at this. What approach would you take?

Exploration vs. Exploitation

Bandit Strategies - Uniform Exploration

Strategy:

- Exploration Phase: Try each arm N times
- Select the arm **a**' with the highest average reward
- Exploitation Phase: Play arm **a**' for the remaining T - N rounds

Bandit Strategies - ε-Greedy

Strategy:

```
For each round $t = 1, 2, \dots$ in $T$:
    Flip a coin with probability $\varepsilon$
    If success:
        Explore: choose an arm uniformly at random
    Else:
        Exploit: Choose the arm with the highest random reward so far
```

Note: A variation exists where you decay ε over time, the assumption being you become more confident in D_a over time

Bandit Strategies - Optimism Under Uncertainty

UCB Algorithm

1) At each round t, we consider two numbers for each arm a

- $N_a(t)$ - the number of times arm a was selected up to round t
- $R_a(t)$ - the sum of the rewards of arm a up to round t

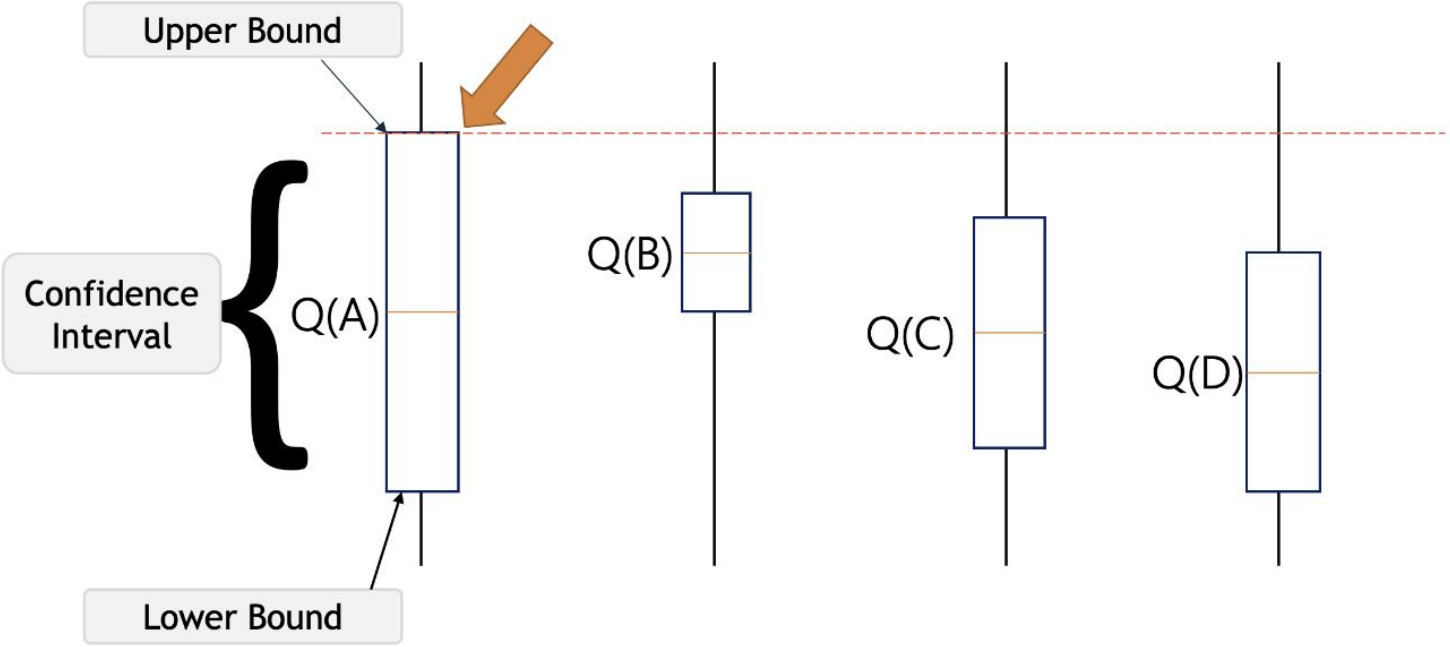
2) From this we compute:

  - The average reward: $r_a(n) = R_a(t) / N_a(t)$
  - The confidence interval: $[r_a(n) - \Delta_a(n), r_a(n) + \Delta_a(n))$, where
    - $\Delta_a(n) = \sqrt{\frac{3\log(t)}{2N_a(t)}}$

3) Select the arm a that has the maximum UCB $r_a(n) + \Delta_a(n)$

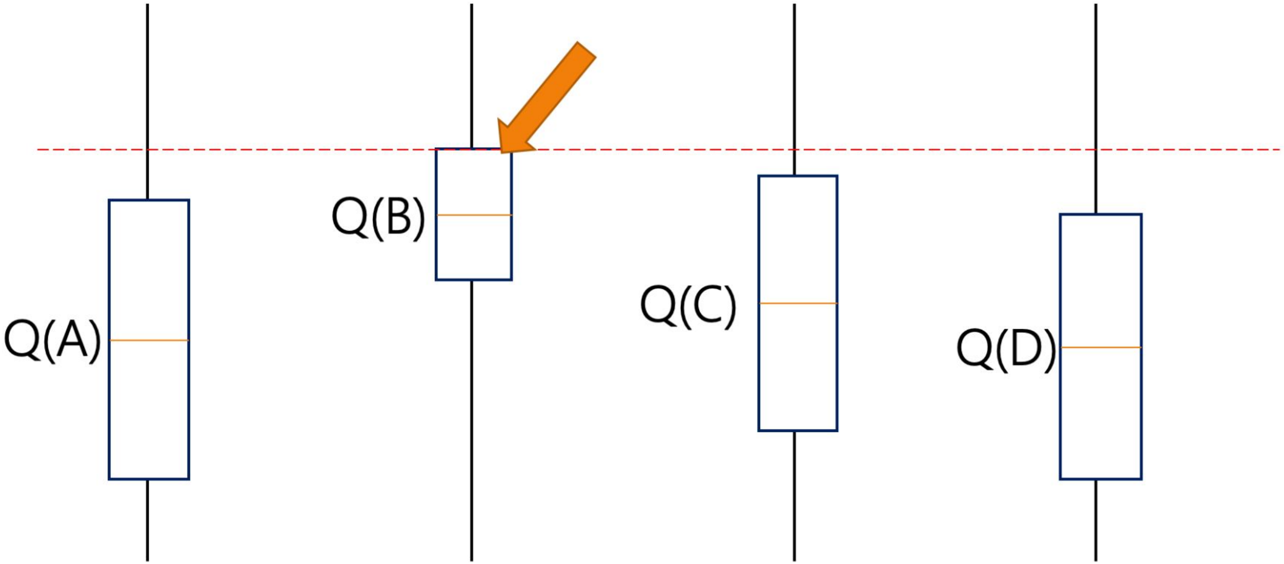
Bandit Strategies - Optimism Under Uncertainty

UCB Algorithm - In Pictures



Bandit Strategies - Optimism Under Uncertainty

UCB Algorithm - In Pictures



Bandit Strategies - Regret

How do we quantify how effective our strategies are?

Regret - the expected reward of always playing an optimal arm compared to the current strategy (i.e how much the algorithm “regrets” not knowing the best arm in advance)

$$R(T) = \mu^* \cdot T - \textstyle\sum_{t=1}^{T} \mu(a_t).$$

Where $\boldsymbol{\mu}\mathbf{(a)} = \mathbf{E(D_a)}$

The “best” mean reward is denoted $\boldsymbol{\mu}^*$ is defined by $\max_{a \in A} \mu(a)$

And $\mathbf{a_t}$ is the actual action taken at time t

Bandits - Recap

Multi-Armed Bandits: Framework for sequential decision-making with multiple actions (arms).

Regret: Measures the missed reward compared to the best action.

Exploration Strategies: Techniques like Uniform Exploration, Epsilon-Greedy, and UCB balance learning and exploitation.

What are some possible extensions?

Extensions to the Bandit Problem

Infinite Arms - Rather than having A arms, what if there were many (infinite?) arms?

Variable Arms - What if D_a changed over time?

Combinatorial Bandits - What if you had to pull multiple arms at a time?

Dueling Bandits - Pull two arms, but rather than being told r_a, you're only told which arm is better?

Contextual Bandits - How do incorporate an external state into your reward expectation?

Extensions to the Bandit Problem

Infinite Arms - Rather than having A arms, what if there were many (infinite?) arms?

Variable Arms - What if D_a changed over time?

Combinatorial Bandits - What if you had to pull multiple arms at a time?

Dueling Bandits - Pull two arms, but rather than being told r_a, you’re only told which arm is better?

Contextual Bandits - How do incorporate an external state into your reward expectation?

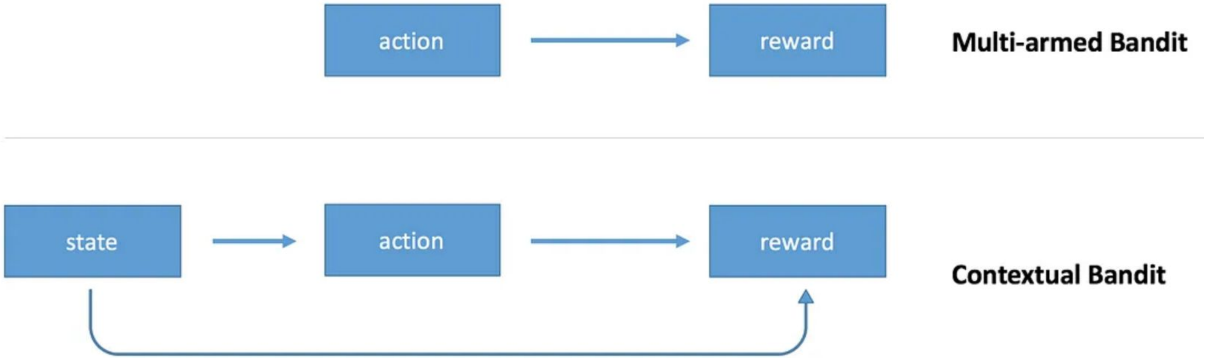
Contextual Bandits

What is a Contextual Bandit?

For t = 1, 2, … , T

1. World presents context as features x_t
   a. Generate reward vector $r_t \in R^K$ (K possible actions)
2. The learning algorithm chooses an action $a_t \in \{1, 2, \dots, K\}$
3. The world presents a reward $r_t(a_t)$ for the action

Optimizes decisions based on previous observations and personalize decisions based on context, which can include human preferences.



Contextual Bandit Application Example

- Personalization of user experience on a website

- Train a model to play chess

- Anything else?

Contextual Bandit Challenges

- Exploitation vs. Exploration
- Computational Efficiency
- Limited feedback from actions
- Need to effectively use context

Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits

- General algorithm for contextual bandits
- Optimal regret bound
  - $\tilde{O}(\sqrt{KT\log|\Pi|})$
- Amortized calls to AMO per round
  - $\tilde{O}(\sqrt{K/T}|)$
- Faster + simpler than predecessors
- Goal: Maximize net reward relative to best policy (minimize regret)
  - Regret is best policy’s average reward - learner’s average reward

$$\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t)) - \sum_{t=1}^{T} r_t(a_t)$$

Arg Max Oracle (AMO)

- Inputs
  - $(x_1, r_1), \ldots, (x_t, r_t)$
- Outputs
  - $$\arg\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t))$$

Algorithm Template

- Start with initial distribution Q_1 over policies Π
- For t = 1, 2, …, T:
    - Observe context x_t
    - Compute distribution p_t over all actions K (using Q_t)
    - Draw action a_t from p_r
    - Collect reward $r_t(a_t)$
    - Compute new distribution Q_{t+1} over policies Π

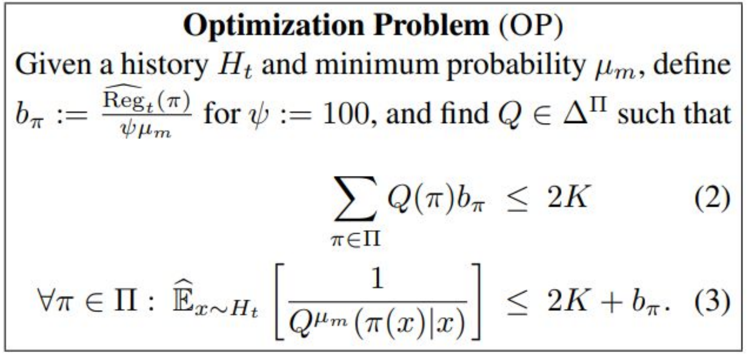
Inverse Probability Weighting

$$\hat{r}_t(a) := \frac{r_t(a_t) \cdot 1\{a = a_t\}}{p_t(a_t)}$$

- Creates reward vector with entries for each action made
- Unbiased estimator for the true reward
- Range/variance bounded → small probabilities cause problems statistically

Optimization Problem (OP)

- Find policy distribution Q satisfying constraints:
    - Low estimated regret (LR)
    - Low estimation variance (LV)
- Theorem: if we obtain policy distributions Q_t via solving the optimization problem, then with high probability, regret after T rounds is at most $\tilde{O}(\sqrt{KT\log|\Pi|})$



Optimization Problem (OP)

Given a history H_t and minimum probability μ_m, define $b_\pi := \frac{\widehat{\mathrm{Reg}}_t(\pi)}{\psi\mu_m}$ for $\psi := 100$, and find $Q \in \Delta^\Pi$ such that

$$\sum_{\pi\in\Pi} Q(\pi)b_\pi \le 2K \tag{2}$$

$$\forall \pi \in \Pi : \ \widehat{\mathbb{E}}_{x\sim H_t}\left[\frac{1}{Q^{\mu_m}(\pi(x)|x)}\right] \le 2K + b_\pi . \tag{3}$$

Coordinate Descent Algorithm to Solve OP

- Input: Initial Weights Q
- Loop:
    - IF LR is violated
        - THEN replace Q by cQ
    - IF there is a policy causing LV to be violated
        - THEN update $Q(\pi) \coloneqq Q(\pi) + \alpha$
    - ELSE
        - Return Q

$0 < c < 1$

$a > 0$

Note: One AMO call is performed per iteration to check violations

Coordinate Descent Complexity

- Õ(√KT / log|Π|) steps for each round of coordinate descent
- Warm starting coordinate descent with the weights computed in the previous epoch
  - Õ(√KT / log|Π|) coordinate descent iterations over all T rounds

Citation:
https://www.youtube.com/watch?v=mi_G5tw7Etg,
https://slideplayer.com/slide/4044385/

K-Armed Dueling Bandits

Motivations

- What are some real-world scenarios where some decisions have better outcomes, but without an explicit reward?
  - Retrieval/search: many methods, but only top-level preferences
  - Recommendation: only observe user's choice amongst options
- How might one quantify the 'goodness' or 'badness' of an outcome without a given quantitative signal?

Framework

- Bandit: element of a ‘strategy set’ – also ‘action’ or ‘arm’
    - Assumption: observations are unbiased estimates of payoff
- When observations are not direct/reliable, use relative comparisons
- K-armed dueling bandits: *minimize regret using noisy comparisons*
- Discrete bandits: more tractable for exploration than a large, continuous search space
- High-level goal: find the **best bandit out of K bandits through exploration** over a fixed horizon. Then **exploit**!

Formal definitions

For given bandits b and b’, we can make pairwise comparisons:

$$P(b > b') = \epsilon(b, b') + 1/2$$

Where ϵ follows some basic properties:

$$\epsilon(b, b') = -\epsilon(b', b) \qquad \epsilon(b, b) = 0 \qquad \epsilon(b, b') \in (-1/2, 1/2)$$

This allows us to probabilistically compare bandits!

Assumption: there is a total ordering on bandits, and b > b’ implies ϵ(b, b’) > 0

Structure of comparison model

Strong stochastic transitivity:

$$b_i \succ b_j \succ b_k \qquad \epsilon_{i,k} \geqslant \max\{\epsilon_{i,j}, \epsilon_{j,k}\}$$

Stochastic triangle inequality:

$$b_i \succ b_j \succ b_k \qquad \epsilon_{i,k} \leqslant \epsilon_{i,j} + \epsilon_{j,k}$$

Both the Bradley-Terry and Gaussian models satisfy these constraints:

$$P(b_i > b_j) = \frac{\mu_i}{\mu_i + \mu_j}$$

$$P(b_i > b_j) = P(X_i - X_j > 0)$$

$$X_i - X_j \sim N(\mu_i - \mu_j, 2)$$

Regret – quantify optimality

Assuming we know the best bandit *b**, we can define regret two ways:

$$R_T = \sum_{t=1}^{T} \max\left\{\epsilon\left(b^*, b_1^{(t)}\right), \epsilon\left(b^*, b_2^{(t)}\right)\right\}$$

Strong regret: fraction of users who prefer *b** over the *worse* of b_1, b_2

$$\tilde{R}_T = \sum_{t=1}^{T} \min\left\{\epsilon\left(b^*, b_1^{(t)}\right), \epsilon\left(b^*, b_2^{(t)}\right)\right\}.$$

Weak regret: fraction of users who prefer *b** over the *better* of b_1, b_2

Algorithmic bookkeeping

Empirical estimate of P(b_i > b_j) after *t* comparisons:

$$\hat{P}_{i,j} = \frac{\#\, b_i\ wins}{\#\ comparisons}$$

Confidence interval on each of P's entries:

$$\hat{C}_t = (\hat{P}_t - c_t,\, \hat{P}_t + c_t) \qquad c_t = \sqrt{4 \log(1/\delta)/t}$$

Interleaved Filter (IF)

```
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 $P(\hat{b} > b)$ according to (6)
$\forall b \in W$, maintain $1-\delta$ confidence interval $\hat{C}_{\hat{b},b}$ of $\hat{P}_{\hat{b},b}$ according to (7), (8)
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. $(\hat{P}_{\hat{b},b} > 1/2 \wedge 1/2 \notin \hat{C}_{\hat{b},b})$ do
        $W \leftarrow W \setminus \{b\}$    $//\hat{b}$ declared winner against $b$
    end while
    if $\exists b' \in W$ s.t. $(\hat{P}_{\hat{b},b'} < 1/2 \wedge 1/2 \notin \hat{C}_{\hat{b},b'})$ then
        while $\exists b \in W$ s.t. $\hat{P}_{\hat{b},b} > 1/2$ do
            $W \leftarrow W \setminus \{b\}$    $//pruning$
        end while
        $\hat{b} \leftarrow b'$,  $W \leftarrow W \setminus \{b'\}$    $//b'$ declared winner against $\hat{b}$ (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 $(\hat{b}, \hat{T})$
```

Initialize parameters, selecting a random ‘best’ bandit. ***W*** is our set of working candidates for the final best bandit.

Update comparison win probabilities & confidence intervals

Prune bandits we are **confident** our best bandit is better than

If there is a bandit we are **confident** is better than our current best:

- remove ones *probably* worse than the current best
- make this bandit the new current best, remove from ***W***
- reset win probabilities & confidence intervals

Results

Returns the correct bandit with P $\geq$ 1 - 1/T

A suboptimal bandit has maximal regret of $\mathcal{O}(T)$

Expected regret is therefore:

$$\mathbf{E}[R_T] \leqslant \left(1 - \frac{1}{T}\right)\mathbf{E}\big[R_T^{IF}\big] + \frac{1}{T}\mathcal{O}(T)$$
$$= \mathcal{O}\big(\mathbf{E}\big[R_T^{IF}\big] + 1\big)$$

Considerations

- Interesting method for finding the best bandit! Especially in the absence of concrete/reliable reward and only given noisy comparisons
- Strong theoretical bounds for mistakes, regret, and number of comparisons required
- Could reasonably be used to ascertain human preferences, e.g. best strategy for recommendations/search using interleaved options

Advancements in Dueling Bandits

Definitions

Condorcet Winner: The bandit who wins a majority of the vote in every head-to-head election against each of the other choices.

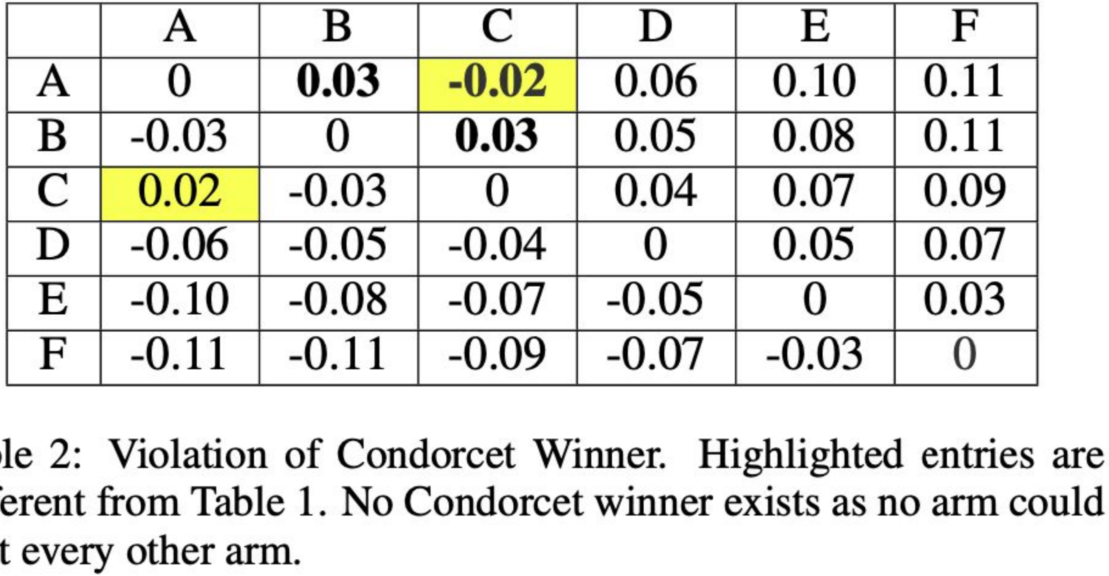
Borda Winner: The bandit with the highest likelihood of winning a duel against a random action.

Von Neumann Winner: A probability distribution W over our bandits such that if we sample a bandit A from W, it will beat a random action with probability greater than 0.5

Sui, Y., Zoghi, M., Hofmann, K., &amp; Yue, Y. (2018). Advancements in dueling bandits. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. https://doi.org/10.24963/ijcai.2018/776

Preference Matrix

A preference matrix describes the probability that any bandit is chosen over other another bandit. In this representation each entry Δ_{JK} is the probability bandit ***J*** is chosen over bandit ***K*** minus 0.5



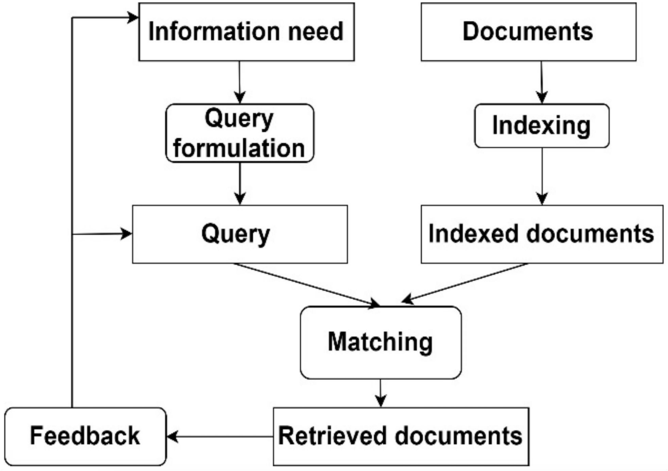
	A	B	C	D	E	F
A	0	**0.03**	**-0.02**	0.06	0.10	0.11
B	-0.03	0	**0.03**	0.05	0.08	0.11
C	0.02	-0.03	0	0.04	0.07	0.09
D	-0.06	-0.05	-0.04	0	0.05	0.07
E	-0.10	-0.08	-0.07	-0.05	0	0.03
F	-0.11	-0.11	-0.09	-0.07	-0.03	0

Table 2: Violation of Condorcet Winner. Highlighted entries are different from Table 1. No Condorcet winner exists as no arm could beat every other arm.

Dueling Bandit Gradient Descent (Over Structured Input Space)

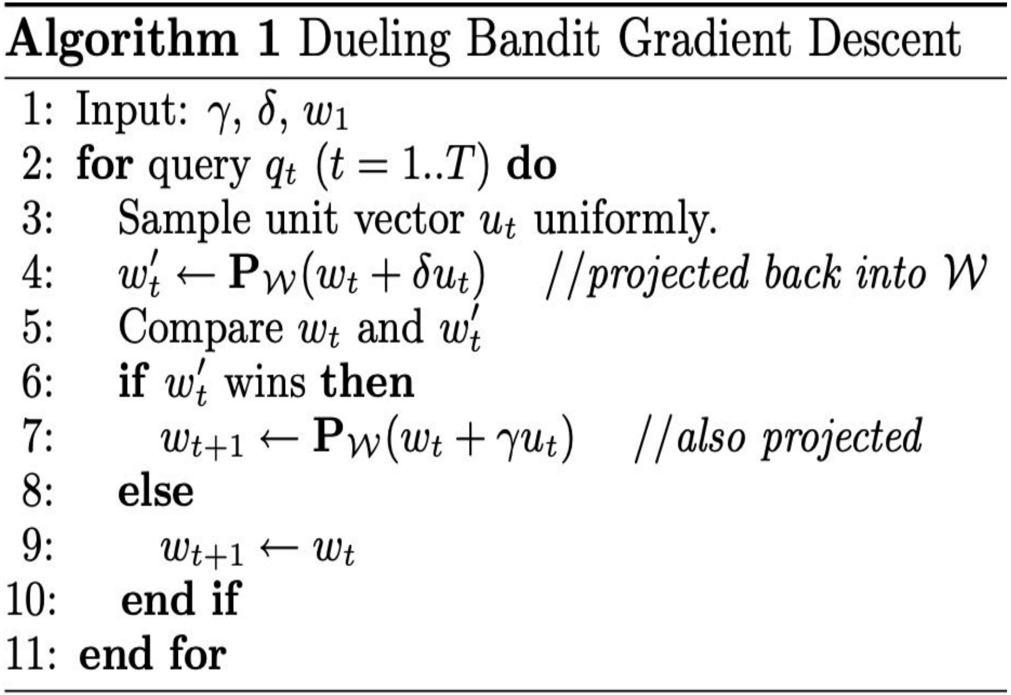
Formulates online retrieval optimization problem as dueling bandits problem over structured input space ***W***, an n-dimensional unit sphere that parametrizes the IR system.

Rather than use comparisons to describe a preference, comparisons are used to compute gradients to find the optimal solution in ***W.***



Yue, Y., & Joachims, T. (2009). Interactively optimizing information retrieval systems as a dueling bandits problem. *Proceedings of the 26th Annual International Conference on Machine Learning*. https://doi.org/10.1145/1553374.1553527

Dueling Bandits Gradient Descent Algorithm



Algorithm 1 Dueling Bandit Gradient Descent

```
Input: $\gamma$, $\delta$, $w_1$
for query $q_t$ $(t = 1..T)$ do
    Sample unit vector $u_t$ uniformly.
    $w'_t \leftarrow \mathbf{P}_{\mathcal{W}}(w_t + \delta u_t)$    //projected back into $\mathcal{W}$
    Compare $w_t$ and $w'_t$
    if $w'_t$ wins then
        $w_{t+1} \leftarrow \mathbf{P}_{\mathcal{W}}(w_t + \gamma u_t)$    //also projected
    else
        $w_{t+1} \leftarrow w_t$
    end if
end for
```

Input learning rate, exploration step size, and starting candidate point in search space

Take random step from current candidate point and duel these points

Update current candidate point if new point was preferred.

Contextual Dueling Bandits

Instead of assuming a fixed preference matrix, contextual dueling bandits assumes a hidden preference matrix $\mathbf{P}_t$ associated with a know context $\mathbf{x}_t$ chosen from some space **X** by nature.

The algorithms in this paper assume a Von Neumann winner, as there is no guarantee on a Condorcet winner for this problem but a Von Neumann winner must exist.

The goal is to choose a Von Neumann winner policy **π** that maps contexts **X** to bandits/actions in action space **A**. That is, a policy that chooses actions that are preferred to random actions more than half the time.

Dudík, M., Hofmann, K., Schapire, R.E., Slivkins, A. &amp; Zoghi, M.. (2015). Contextual Dueling Bandits. <i>Proceedings of The 28th Conference on Learning Theory</i>, in <i>Proceedings of Machine Learning Research</i> 40:563-587 Available from https://proceedings.mlr.press/v40/Dudik15.html.

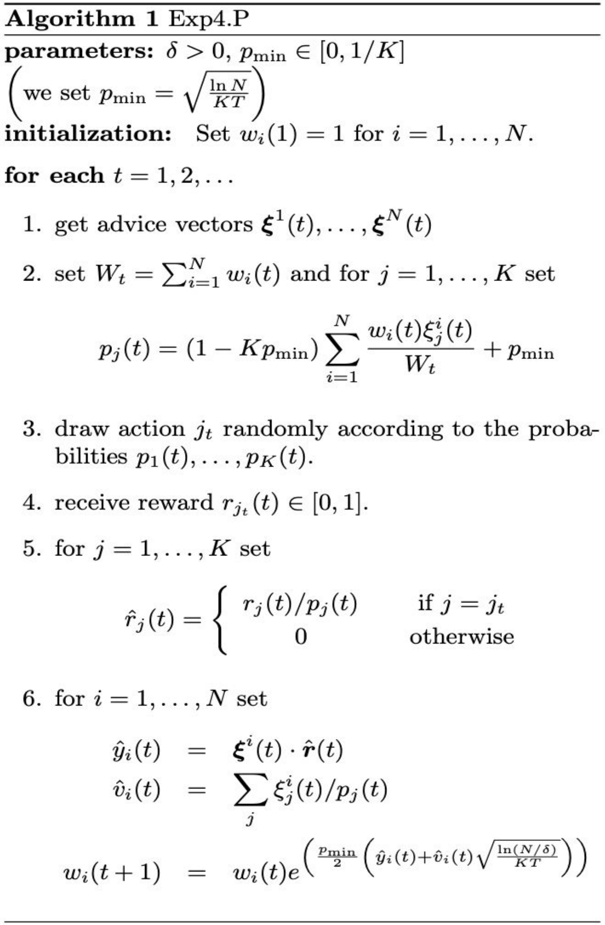
Sparring Algorithms

While some dueling bandit algorithms sample both points to duel against each other, others formulate the problem as two traditional bandit algorithms each tasked with maximizing the probability of its choice being preferred over its adversary.

Contextual dueling bandits involves sparring ‘meta-duels’, where at each time step, our adversaries will choose a policy $\boldsymbol{\pi}$ mapping context space ***X*** to action space ***A***. We will then sample from ***X*** and duel the choices made by the adversaries from their proposed policies.

Dudik et al. proposes dueling two EXP.4 algorithms against each other.

Sparring EXP.4 for Contextual Dueling Bandits



```
Algorithm 1 Exp4.P
parameters: $\delta > 0$, $p_{\min} \in [0, 1/K]$
    $\left(\text{we set } p_{\min} = \sqrt{\frac{\ln N}{KT}}\right)$
initialization: Set $w_i(1) = 1$ for $i = 1, \ldots, N$.
for each $t = 1, 2, \ldots$
    1. get advice vectors $\boldsymbol{\xi}^1(t), \ldots, \boldsymbol{\xi}^N(t)$
    2. set $W_t = \sum_{i=1}^N w_i(t)$ and for $j = 1, \ldots, K$ set
           $p_j(t) = (1 - Kp_{\min}) \sum_{i=1}^{N} \frac{w_i(t)\xi_j^i(t)}{W_t} + p_{\min}$
    3. draw action $j_t$ randomly according to the probabilities $p_1(t), \ldots, p_K(t)$.
    4. receive reward $r_{j_t}(t) \in [0, 1]$.
    5. for $j = 1, \ldots, K$ set
           $\hat{r}_j(t) = \begin{cases} r_j(t)/p_j(t) & \text{if } j = j_t \\ 0 & \text{otherwise} \end{cases}$
    6. for $i = 1, \ldots, N$ set
           $\hat{y}_i(t) = \boldsymbol{\xi}^i(t) \cdot \hat{\boldsymbol{r}}(t)$
           $\hat{v}_i(t) = \sum_j \xi_j^i(t)/p_j(t)$
           $w_i(t+1) = w_i(t) e^{\left(\frac{p_{\min}}{2}\left(\hat{y}_i(t) + \hat{v}_i(t)\sqrt{\frac{\ln(N/\delta)}{KT}}\right)\right)}$
```

Exp4 Summary: We parametrize our bandits as vectors. When we observe the context we create a probability distribution over our bandits based where each bandits probability is a function of its vectors dot product with the current context. Exp4 chooses its bandit by sampling from this probability distribution.

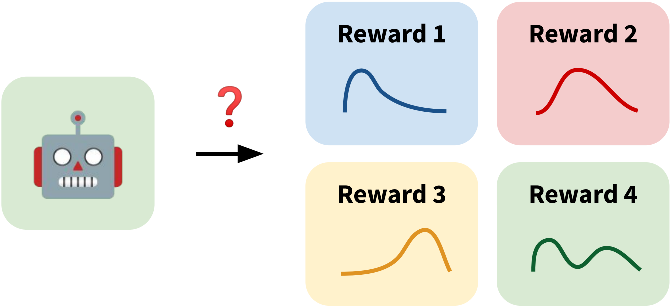
We then duel the choices of our two Exp4 algorithms and propagate the observed reward from this duel back through their representation vectors.

Applications

Recap

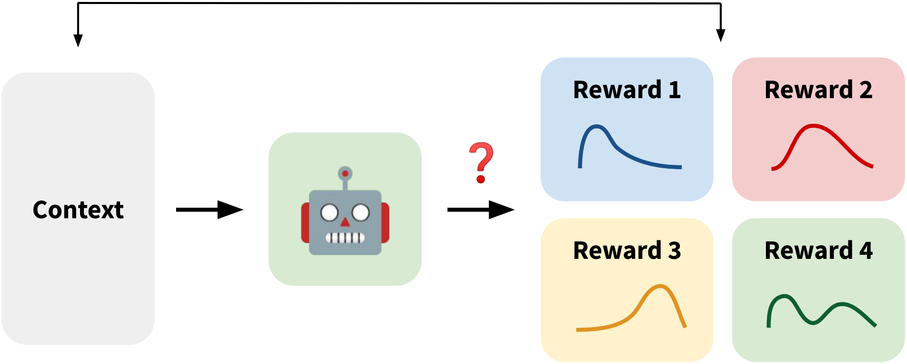
Multi-armed Bandit (MAB)

- Given K arms each having a fixed reward distribution
- Learn to choose its actions to maximize the cumulative rewards over time



Contextual Bandit

- Given K arms and an observed context
- Agent receive only a reward sampled from the chosen arm
- Learn the relationship between context and rewards to maximize the cumulative rewards



Real-life applications of Bandit

Healthcare

- **Warfarin dosing.** Too low or high initial dosage can result in stroke or internal bleeding. [1] model contextual bandit problem to assign appropriate dosage to patients.
- **Brain and behavioral modeling.** [2] extend Thompson Sampling to study reward-processing biases associated with different mental disorders.

Dynamic Pricing

- [5] propose to combine MAB with partial identification of consumer demand from economic theory to derive algorithm that balances short and long term profit.

Finance

- **Online portfolio optimization.** [3] propose a bandit algorithm for making online portfolio by utilizing an upper confidence bound bandit framework. [4] incorporate risk-awareness into the classic MAB setting.

Recommendation Systems

- [6] propose large-scale contextual bandit strategies that continually explore to mitigate the cold start problem in recommender systems.
- [7] propose a Freshness Aware Thompson Sampling that manages the recommendation of fresh document according to the user's risk of the situation.

1. Bastani and Bayati, Online decision-making with high-dimensional covariates, 2015
2. Bouneffouf et al., Bandit models of human behavior: Reward processing in mental disorders, 2017
3. Shen et al., Portfolio Choices with Orthogonal Bandit Learning, 2015
4. Huo and Fu, Riskaware multi-armed bandit problem with application to portfolio selection, 2017
5. Misra et al., Dynamic online pricing with incomplete information using multi-armed bandit experiments, 2018
6. Zhou et al., Large-scale bandit approaches for recommender systems, 2017
7. Bouneffouf, Freshnessaware thompson sampling, 2014.

Real-life applications of Bandit

Dialogue Systems

- **Dialogue response selection.** [1] propose contextual MAB with non-linear reward function that uses distributed representation of text for online response selection.
- **Pro-activity dialogue selection.** [2] propose Contextual Attentive Memory Network which generalize contextual bandit to attend temporal information which is the past interactions of the agent.
- **Multi-domain dialogue selection.** [3] attempts online posterior dialogue orchestration, defining it as the process of selecting the most suitable subset of skills in response to a user’s input.

Anomaly Detection

- [4] study anomaly detection in interactive setting. The goal is to maximize the number of true anomalies confirmed by the human expert, adhering to a predefined query budget. They employ a multi-armed bandit framework to develop a collaborative contextual bandit algorithm.

1. Liu et al., Customized nonlinear bandits for online response selection in neural conversation models, 2018
2. Silander et al., Model-independent online learning for influence maximization, 2017
3. Upadhyay et al., A bandit approach to posterior dialog orchestration under a budget, 2018
4. Ding et al., Interactive anomaly detection on attributed networks, 2019