Multi-armed Bandit Literature Review

Lisa Liu
Dept. of Computer Science
Princeton University
xl2493@princeton.edu

Jane Pan
Dept. of Computer Science
Princeton University
jp7224@princeton.edu

Nikhil Pimpalkhare
Dept. of Computer Science
Princeton University
np6641@princeton.edu

Abstract

This paper surveys a range of online algorithmic approaches to the multi-armed bandit problem. We introduce and define the multi-armed bandit problem, as well as its practical applications. We present a formal overview of several classic online algorithms for this problem, including epsilon-greedy, upper confidence bound, and Thompson Sampling. We then describe three modern state-of-the-art approaches and applications of the multi-armed bandit problem which incorporate causal inference, Bayesian statistics, and combinatorial decomposition into their frameworks. We also provide analysis on the regret for some of the algorithms, as well as showing the advantage and drawback of them.

1 Introduction

The multi-armed bandit problem is a cornerstone problem of reinforcement learning. Conceived in the 1950s, this classic stochastic scheduling problem models a learner that simultaneously attempts to explore the environment and exploit existing knowledge to optimize their decisions. The agent attempts to balance the trade-off between these competing tasks to maximize their total value over the period of time.

In this paper, we will formally define the problem of multi-armed bandit, along with necessary terminology. Then we will present various classic and modern algorithms for this problem.

1.1 Problem Definition

Robbins (1952) was the first to introduce an early variant of multi-armed bandits, which Robbins termed "a problem of two populations":

Definition 1 (A Problem of Two Populations) *Given two statistical populations with unknown univariate cumulative distribution functions and a fixed number of sample draws, how should we draw* $x_1, ..., x_n$ *to maximize* $S_n = \sum_{i=1}^{n} x_i$*?*

Unlike other statistical problems, Robbins (1952) emphasized that the two-population problem's goal is to decide how to draw the sample, rather than merely estimating each population's distribution.

Today, the multi-armed bandit problem is often introduced with the following thought experiment: consider a gambler who has access to k slot machines. In a given round t, he chooses to play any of the machines, i.e. *arms*. Let us assume he chooses the ith arm; then he observes a reward r_{it}, which is drawn from the arm's reward distribution R_i (which is unknown to the bandit). His goal is to maximize the total reward across a fixed n rounds.

We can describe a generic multi-armed bandit algorithm thusly:

Algorithm 1 Multi-Armed Bandit Algorithm Protocol

1: Given: k arms, horizon of T, and arm reward distributions $R_{i\,i=1}^{k}$
2: **for** t=1, . . . , T **do**
3: Algorithm selects arm a_t
4: Reward r_t is sampled from R_t
5: Algorithm collects r_t but does not see other rewards.
6: **end for**

It is assumed that successive pulls of the same arm are independently and identically distributed according to the unknown reward distribution R_i and with unknown expectation μ_i. Moreover, it is also assumed that rewards across different machines are independently distributed. Also, we consider only the formulation where the *bandit feedback*, i.e. the information observed by the agent, is limited to the reward of the arm it selected (and no other arm).

The goal of the problem is to develop an optimal policy π (i.e. allocation strategy) so as to maximize the total expected reward.

1.1.1 Useful Terminology

Here we define some useful terms applicable to the multi-armed bandit problem.

Definition 2 (Regret) *The regret after T rounds is defined as the difference between the optimal reward sum and the sum of the collected rewards, i.e.:*

$$\rho(T) = T\mu^* - \sum_{t=1}^{T} r_{it}$$

where

$$\mu^* = max_{1 \leq i \leq k} \mu_i$$

or can be rewritten as

$$\rho(T) = \max_{1 \leq i \leq K} \sum_{t=1}^{T} (\mu^* - \mu_i)$$

Definition 3 (Zero-regret strategy) *A zero-regret strategy is one whose average regret per round tends to zero with certain probability as the time horizon goes to infinity. That is to say, with probability 1, a zero-regret strategy π has:*

$$lim_{T \to \infty} \frac{\rho_\pi}{T} = 0 \tag{1}$$

Definition 4 (Horizon) *The number of rounds left to play.*

Definition 5 *We define an algorithm to solve multi-armed bandit problem if it can match a lower bound, $\rho(T) = O(\log T)$.*

This is from a classical result from Lai and Robbins (1985) that

$$\lim_{T \to \infty} \frac{\mathbb{E}[\text{Regret}(T)|\mu]}{\log(T)} = \sum_{k \neq k^*} \frac{\mu_{k^*} - \mu_k}{d_{\text{KL}}(\mu_{k^*}||\mu_k)}$$

where $d_{\text{KL}}(\theta||\theta')$ is the Kullback-Leibler divergence. Thus, no matter how good an algorithm for solving the multi-armed bandit problem is, its expected regret grows at least logarithmically.

1.1.2 The Multi-Armed Bandit in the Online Setting

The multi-armed bandit problem is inherently online. The agent has no information of the reward distributions of any of the machines when the game begins and only iteratively accrues information as it pulls a lever and learns the reward for each round of the game. While some strategies apply a batched offline approach (using batches of historical data to construct the strategy for the next batch of incoming data), we consider online algorithms only. In most real world applications of the problem, once a player has moved on from a choice they may never visit it again, which is why online algorithms are perfectly applicable.

1.2 Applications

The multi-armed bandit problem has many real life applications. As introduced in Slivkins (2021), the original motivation was meant to design ethical medical trials to minimize the pain of patients while collecting more useful scientific data. The choice in this case will be to decide which drug to prescribe and the reward will be health for the patients.

More recent and modern applications typically involve user behavior on the Internet. For example, deciding which advertisement to place on the webpages is a huge application nowadays. Displaying and recommending the right advertisement can incur much more revenue from the advertisements. For many pages, user will never return to a page they have visited. Therefore, the webpage owner will have to decide which advertisement to display to attract more users and draw more revenue in that single visit. Another common application would be financial portfolio design. However, the difference would be that the feedback in this case, the previous prices for all stocks, will be publicly available no matter which action you take. The strategies for different applications of multi-armed bandit problem will therefore vary and depend on many different aspects of the problem.

1.3 Exploration vs. Exploitation

At the heart of any algorithmic approach to the multi-armed bandit problem is the trade-off between exploratory strategies which aim to increase the agent's information about the reward distributions R_t defining the current problem instance, and exploitative strategies which leverage all information the algorithm has gained about the arms to earn as much reward as possible. Initially, exploration is extremely important to prevent premature convergence to a suboptimal exploitation strategy. However, exploring for too long can lead to an unnecessary increase in regret if the information that we learn does not change the decisions we make during exploitation.

Exploitation strategies are usually quite standard regardless of the algorithm used. In most cases, the algorithm keeps some probability distribution per arm and simply selects one greedily when it is time to exploit. Some examples of greedy selection are the lowest mean reward of the seen samples per arm and the lowest sample from the aforementioned probability distribution. Then, algorithms are differentiated in the way in which they approach two subproblems. Firstly, which arms should an algorithm select in order to gain the most useful information for future decision? Secondly, when should the algorithm switch from exploration to exploitation?

2 Classic Approaches

2.1 Epsilon-Greedy

Perhaps the best known of the classical approaches, this class of algorithms explicitly addresses the exploitation-exploration paradigm by performing exploration with probability ϵ and exploitation otherwise.

2.1.1 Original ϵ-Greedy Approach

First described in Watkins (1989), this approach takes in a fixed exploration probability ϵ and otherwise selects the arm with the highest empirical mean.

Algorithm 2 ϵ-Greedy Algorithm

Given: fixed $\epsilon \in (0, 1)$
For each arm $k = 1, \ldots, K$, set $\hat{\mu}_k = 0$
for t=1,2,... **do**
    With probability ϵ select a random arm; else, choose a_k such that $k = argmax_k \hat{\mu}_k$
    Play arm k and observe reward r_k
    Update $\hat{\mu}_k$
end for

Theorem 6 *The ϵ-greedy algorithm (with a constant ϵ) has linear regret.*

Proof. Suppose that the initial estimate of the reward distribution means are perfect, i.e. that a_k such that $k = argmax\hat{\mu}_k$ is indeed the optimal arm to draw. Then with probability ϵ the arm selects a suboptimal arm. Thus the expected regret is ϵT, i.e. linear. ■

2.1.2 ϵ-First Algorithm

Developed by Even-Dar et al. (2006), this algorithm is centered on the idea that while we may not be able to always select the best arm for exploitation, we would like to at least select an arm with an expected reward that is not too far off from optimal.

Definition 7 (α-Optimal Arm) *An arm a is α-optimal if its expected reward is at most α below the optimal reward, i.e.:*

$$E[r_{at}] \geq r_t^* - \alpha$$

In this algorithm, the agent spends the first $\epsilon \cdot T$ rounds in exploration. The remaining rounds are spent in exploitation, where the agent pulls only the arm with the highest empirical mean found in the exploration phase. Ideally, with sufficient time spent in exploration, the agent will choose an α-optimal arm (i.e. one that is not too much worse than the optimal arm).

Algorithm 3 ϵ-First Algorithm

Given: fixed $\epsilon \in (0, 1)$
For each arm $k = 1, \ldots, K$, set $\hat{\mu}_k = 0$
for t=1,..., $\epsilon \cdot$ T **do**
    Choose a random arm a_i
    Play arm i and observe reward r_i
    Update $\hat{\mu}_i$
end for
for t=$\epsilon \cdot$ T+1,..., T **do**
    Choose a_k such that $k = argmax_k \hat{\mu}_k$
    Play arm k and observe reward r_k
end for

Theorem 8 *In the exploration phase of the algorithm, a sample complexity of $O(\frac{K}{\alpha^2} log(\frac{K}{\delta})$ produces an $\alpha - optimal$ arm with probability at least $1 - \delta$.*

Proof. Consider a non-α-optimal arm a' (i.e. $E[R(a)] < r^* - \alpha$). We want to show that the probability that algorithm chooses a' over the optimal arm $a*$ is bounded by δ. The algorithm will choose a' if it finds that $\hat{\mu}_{a'} > \hat{\mu}_{a*}$. Applying the Hoeffding inequality gives:

$$\begin{aligned} P[\hat{\mu}_{a'} > \hat{\mu}_{a*}] &\leq P[\hat{\mu}_{a'} > E(R(a')) + \frac{\epsilon}{2}] + P[\hat{\mu}_{a'} < E(R(a')) - \frac{\epsilon}{2}] \\ &\leq 2e^{-2(\frac{\epsilon}{2})^2 l} \end{aligned}$$

Now let $l = \frac{2}{\epsilon^2} ln(\frac{2K}{\delta})$. Plugging this into the last line:

$$P[\hat{\mu}_{a'} > \hat{\mu}_{a*}] \leq 2e^{-2(\frac{\epsilon}{2})^2 \cdot \frac{2}{\epsilon^2} ln(\frac{2K}{\delta})}$$
$$= \frac{\delta}{K}$$

Thus the probability that we select a particular non-optimal arm is upper bounded by $\frac{\delta}{K}$. Taking the union bound over all possible non-optimal arms gives us:

$$(K-1)\frac{\delta}{K} < \delta$$

■

2.1.3 ϵ-decreasing

So far, we have only considered strategies with a fixed ϵ, i.e. a fixed allocation of exploration phase. Cesa-Bianchi and Fischer (1998) introduced the ϵ-decreasing algorithm, in which we decrease ϵ so as to allow the algorithm to get arbitrarily close to the optimal strategy as the time horizon approaches infinity. The intuition behind this algorithm is that, with a constant ϵ, even if we choose the optimal strategy with probability $1-\epsilon$, we will always waste time trying a non-optimal arm with ϵ probability. However, as our time horizon grows asymptotically large, we will have increasingly better estimates of the reward distributions of the arms and are therefore more likely to be correctly choosing the optimal arm in the exploitation phase. Therefore, as we play more and more rounds, we should reduce the amount of time we spent trying random arms and focus on the arms that we are reasonably sure are optimal. Therefore, we gradually decrease ϵ so that as T goes to infinity, ϵ approaches zero. Hence, as our time horizon approaches infinity, our ϵ-greedy algorithm can approach the optimal strategy.

Algorithm 4 ϵ-Decreasing Algorithm

1: Given: fixed $\epsilon_0 \in (0,1)$
2: Initialize $\epsilon_i{}_{i=1}^{T}$ such that each $\epsilon_i = min\{1, \epsilon_0 \frac{log(t)}{t}\}$
3: For each arm $k = 1, \ldots, K$, set $\hat{\mu}_k = 0$
4: **for** t=1,..., T **do**
5: With probability ϵ_t select a random arm; else, choose a_k such that $k = argmax_k \hat{\mu}_k$
6: Play arm k and observe reward r_k
7: Update $\hat{\mu}_k$
8: **end for**

2.2 UCB

The Upper Confidence Bound (UCB) algorithm introduced in Auer et al. (2002) is a direct application of the exploration versus exploration tradeoff in multi-armed bandit algorithms. This algorithm makes an important constraint on the problem formulation - each arm is associated with a Bernouilli (coin flip) distribution, and thus the rewards are always either zero or one.

For each round, this algorithm assigns a score to each arm based off of the mean seen reward thus far from that arm and the number of times that arm has been sampled thus far. At each iteration, the algorithm is simply to pick the arm with the highest score. Concretely, the score of arm i after t rounds is:

$$UCB_t(i) = \hat{\mu_{t,i}} + \sqrt{\frac{\ln(t)}{n_{t,i}}}$$

In the above equation, $\hat{\mu_{t,i}}$ is the mean of all of the rewards we have seen in previous rounds after having selected arm i. $n_{t,i}$ is the number of times which we have selected arm i thus far.

The first term, $\hat{\mu_{t,i}}$, indicates a clear exploitative strategy: we want to pick an arm which has resulted in high reward in the past. The second term, $\sqrt{\frac{\ln(t)}{n_{t,i}}}$, forces the algorithm to also explore; the term keeps increasing as we play more rounds, and we will have to occasionally visit every arm in order to keep the term down and allow the algorithm to exploit.

2.2.1 Analysis

Much of this analysis is inspired by lecture notes by Agrawal (2019). The structure of the exploration term in the definition $UCB_i(t)$ has a profound connection to the Chernoff Bound which allows this algorithm to have excellent theoretical guarantees for expected regret.

Lemma 9 *If $\hat{\mu_{t,i}}$ is the calculated mean of the rewards from selecting arm i seen after t rounds and μ_i is the true mean reward received from selecting arm i, $P(|\mu_i - \hat{\mu_{t,i}}| \geq \sqrt{\frac{\ln(t)}{n_{t,i}}}) \leq \frac{2}{t^2}$.*

Proof. Since we have constrained our rewards to come from Bernouilli random variables, we can apply a corollary from the Chernoff Bound called Hoeffding's Inequality which holds for the mean of i.i.d Bernouilli variables. It states:

$$P(|\bar{X} - E[\bar{X}]| \geq \delta) \leq 2e^{-2\delta^2 n}$$

Applying this to our calculated mean with $\delta = \sqrt{\frac{\ln(t)}{n_{t,i}}}$, we immediately get the result we want:

$$P(|\mu_i - \hat{\mu_{t,i}}| \geq \sqrt{\frac{\ln(t)}{n_{t,i}}}) \leq 2e^{-2(\frac{\ln(t)}{n_{t,i}})n_{t,i}} = 2t^{-2}$$

■

Intuitively, this lemma means that that the exploration term is an high probability upper bound on the error of $\hat{\mu_{t,i}}$. We can use this fact to make a powerful claim about the probability that we make a suboptimal selection:

Lemma 10 *At any point t, given that we have visited a suboptimal arm $n_{t,i} \geq \frac{4\ln(t)}{(\mu^* - \mu_i)^2}$ times, the probability of selecting i is less than or equal to $\frac{4}{t^2}$.*

Proof. Let i^* be the correct optimal choice. The only way we pick a suboptimal arm i is if $UCB_{t,i} \geq UCB_{t,i^*}$.

$$UCB_{t,i} = \hat{\mu_{t,i}} + \frac{4\ln(t)}{(\mu^* - \mu_i)^2} \leq \hat{\mu_{t,i}} + \frac{\mu^* - \mu_i}{2}$$

given that $n_{t,i} \geq \frac{4\ln(t)}{(\mu^* - \mu_i)^2}$.

Now, using Lemma 9, given that $n_{t,i} \geq \frac{4\ln(t)}{(\mu^* - \mu_i)^2}$, we know that $|\mu_i - \hat{\mu_{t,i}}| < \sqrt{\frac{\ln(t)}{n_{t,i}}} \leq \frac{\mu^* - \mu_i}{2}$ with probability at least $1 - \frac{2}{t^2}$. Then, adding $|\mu_i - \hat{\mu_{t,i}}|$ to both sides of the expression above, we get

$$UCB_{t,i} \leq \mu_i + (\mu^* - \mu_i) = \mu^*$$

Now, applying Lemma 9 to i^*, we get $|\hat{\mu_{t,i^*}} - \mu^*| < \sqrt{\frac{\ln(t)}{n_{t,i^*}}}$ with probability at least $1 - \frac{2}{t^2}$. Then, algebraicly, $UCB_{t,i^*} = \hat{\mu_{t,i^*}} + \sqrt{\frac{\ln(t)}{n_{t,i^*}}} > \mu^*$.

Thus, we have $UCB_{t,i} < UCB_{t,i^*}$ with probability at least $1 - \frac{4}{t^2}$ by union bound. If this is the case, we will not pick suboptimal arm i. Then, the probability of selecting i is less than or equal to $\frac{4}{t^2}$. ■

This leads to the following upper bound on the expected number of pulls of any given suboptimal arm i:

Lemma 11 *For any suboptimal arm i,* $E[n_{t,i}] \leq \frac{4\ln(t)}{\mu^* - \mu_i} + 8$.

Proof.

$$E[n_{t,i}] \leq \frac{4\ln(t)}{(\mu^* - \mu_i)^2} + E[n_{t,i} | n_{t,i} \geq \frac{4\ln(t)}{(\mu^* - \mu_i)^2}]$$

$$E[n_{t,i}] \leq \frac{4\ln(t)}{(\mu^* - \mu_i)^2} + \sum_t \frac{4}{t^2} \leq \frac{4\ln(t)}{(\mu^* - \mu_i)^2} + 8$$

$$E[n_{t,i}] \leq \frac{4\ln(t)}{\mu^* - \mu_i} + 8$$

■

This leads to the following theorem:

Theorem 12 *Using the UCB algorithm,* $E[Regret] \leq \sum_i (4\ln(T) + 8(\mu^* - \mu_i))$.

Proof.

$$E[Regret] = \sum_i (E[n_{T,i}] * (\mu^* - \mu_i)) \leq \sum_i (4\ln(T) + 8(\mu^* - \mu_i))$$

by direct application of Lemma 11 ■

This result is quite a bit better than the regret guarantee of the epsilon-greedy algorithm. The UCB algorithm achieves logarithmic regret, an order of magnitude better than the previous algorithm. In fact, according to Definition 5, this performance is the best we can hope for.

2.3 Thomson Sampling

Thomson Sampling is the first algorithm for the problem, proposed by Thompson (1933) in 1933 for allocating experimental effort in two-armed bandit problems arising in clinical trials Russo et al. (2017). The algorithm was largely ignored until recently when Chapelle and Li (2011) displayed its strong empirical performance.

2.3.1 Thompson Sampling for Bernoulli Bandits

Thompson (1933) mostly focused on Bernoulli bandits with two arms, and the algorithm is mostly applied with the Bayesian assumption to stochastic bandits, such that the probability of success for each arm is drawn from some fixed distribution. Therefore we will present the algorithm for Bernoulli bandits and discuss some extensions to the general stochastic bandit problem. We will start with the definition of Bernoulli bandit.

Definition 13 (Bernoulli Bandit) *Suppose there are* K *actions, and when played, any action yields either a success or a failure. Each action* $k \in \{1, \ldots, K\}$ *produces a success with probability* $\theta_k \in [0, 1]$*, which is unknown to the agent, but fixed over time. The reward for each success is 1 and 0 otherwise.*

Therefore, from the definition of the Bernoulli Bandit, we can see the success probabilities $(\theta_1, \ldots, \theta_K)$ can be estimated through experimentation. The algorithm for Bernoulli bandits maintains Bayesian priors on the Bernoulli means θ_k's. Thus the reward can be conveniently modeled with Beta distribution since the Beta distribution is the conjugate prior for Bernouilli random variables.

The Thompson Sampling algorithm will assume an initial prior of Beta(1,1) for all θ_k, which is the uniform distribution on (0,1). At each time step t, after having observed $S_k(t)$ successes with reward 1 and $F_k(t)$ failures with reward 0 in $r_k(t) = S_k(t) + F_k(t)$ rounds of playing arm k, the algorithm will update the distribution on θ_k as Beta$(S_k(t) + 1, F_k(t) + 1)$. The algorithm will then sample from the posterior distributions for θ_k's and plays the arm with maximum probability of its θ_k being the largest. The algorithm can be summarized as follows:

Algorithm 5 Thompson Sampling for Bernoulli Bandits

For each arm $k = 1, \ldots, K$, set $S_k = 0, F_k = 0$
for t=1,2,... **do**
  for k=1,..., K **do**
    Sample $\hat{\theta}_k \sim$Beta$(S_k + 1, F_k + 1)$
  end for
  $k \leftarrow \arg\max_k \theta_k$
  Play arm k and observe reward r_k
  $(S_k, F_k) \leftarrow (S_k + r_k, F_k + 1 - r_k)$
end for

2.3.2 Thompson Sampling for General Stochastic Bandit

The extension of Thompson Sampling to the general stochastic bandit is very similar to that for Bernoulli Bandits. The algorithm will draw a random sample from the prior distribution p, and apply the actions that maximize the expected reward. If there are a finite set of possible observations y_t, the expectation will be

$$\mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x] = \sum_o q_{\hat{\theta}}(o|x)r(o)$$

Then the distribution p is updated by conditioning on the realized observation $\hat{y}_t$. If θ is restricted to values from a finite set, then by Bayes rule, this conditional distribution will be

$$\mathbb{P}_{p,q}(\theta = u|x_t, y_t) = \frac{p(u)q_u(y_t|x_t)}{\sum_v p(v)q_v(y_t|x_t)}$$

Therefore we can summarize the algorithm as follows:

Algorithm 6 Thompson Sampling for General Stochastic Bandit

for t=1,2,... **do**
  Sample $\hat{\theta} \sim p$
  $x_t \leftarrow \arg\max_{x \in \mathcal{X}} \mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x]$
  Play arm x_t and observe reward y_t
  $p \leftarrow \mathbb{P}_{p,q}(\theta \in \cdot|x_t, y_t)$
end for

Theorem 14 *For the N-armed stochastic bandit problem, Thompson Sampling algorithm has expected regret*

$$\mathbb{E}[\mathcal{R}(T)] \leq O\left(\left(\sum_{a=2}^{N} \frac{1}{\Delta_a^2}\right)^2 \ln n\right)$$

in time T, where $\Delta_i = \theta^* - \theta_i$, and θ^* is the unique optimal arm. Agrawal and Goyal (2011)

Proof. We will present the intuition of the proof; for the full details, the original proof is available in Agrawal and Goyal (2011).

At any step T, we divide the set of suboptimal arms into two subsets: saturated and unsaturated. The set $C(t)$ of saturated arms at time t is the set of arms a that have already been played a sufficient number of times ($L_a = 24(\ln T)/\Delta_a^2$)) so that with high probability, $\hat{\theta}_a(t)$ is tightly concentrated around θ_a. Then we can estimate the number of steps between two consecutive plays of the optimal arm. After jth play, the $(j+1)$th play of the optimal arm k will occur at the earliest time t such that $\theta_k(t) > \theta_i(t), \forall i \neq k$. We can approximate the number of steps before $\theta_k(t)$ is greater than $\theta_a(t)$ of all saturated arms $a \in C(t)$ using a geometric random variable. However, even if $\theta_k(t) > \theta_a(t)$ for all saturated arms $a \in C(t)$, it may not get played due to play of an unsaturated arm u with a greater $\theta_u(t)$. Call this event an "interruption" by unsaturated arms. It can be shown that if there have been j plays of the optimal arm with $s(j)$ successes, the expected number of step until the $(j+1)$th play can be upper bounded by the product of the expected value of the geometric random variable, and the number of interruptions by the unsaturated arms. The toal number of interruptions by unsaturated

arms is bounded by $\sum_{u=2}^{N} L_u$ since an arm u becomes saturated after L_u plays. Then we can derive the bound assuming the worst case allocation of these $\sum_u L_u$ interruptions. ■

Note that the major difference between this extended general Thompson Sampling and the Bernoulli Thompson Sampling is how we calculate the expected distribution and how we update the distribution given the observed output.

2.3.3 Advantage and Limitations of Thompson Sampling

The reason that Thompson Sampling works is because as information gets collected, beliefs about action rewards are carefully tracked and updated. By sampling actions according to the updated posterior probability that are optimal given existing experiments, the algorithm still continues to sample all actions that could be optimal, and discarding those that are unlikely to be optimal.

However, Thompson sampling does have some limitations that make it not suitable for certain problems. For problems that do not require exploration, it is usually outperformed by greedier algorithms that do not spend so much time exploring. In addition, for time-sensitive problems, Thompson sampling might not be as appropriate as algorithms that exploit a high performing suboptimal action. And for problems requiring careful assessment of information gain, Thompson sampling's strategy of testing the most promising actions might not be optimal as well.

3 Current Approaches

In this section, we provide an overview of three state-of-the-art approaches and applications of the multi-armed bandit problem.

3.1 Doubly-Adaptive Thompson Sampling

Dimakopoulou et al. (2021b) present a variant on Thompson sampling that utilizes causal inference to adaptively reweight the terms of a doubly-robust estimator of the expected reward for each arm. Because the arms are adaptively selected in both Thompson sampling and UCB, the sample averages of the arm rewards are biased and nonnormal. Standing approaches which attempt to debias these estimates do so at the cost of failing to satisfy the variance convergence property; thus the central limit theorem does not apply to them and they also are not asymptotically normal.

3.1.1 Previous Work

The Doubly-Adaptive Thompson Sampling (DATS) algorithm builds off of several previous works. We provide some useful definitions and background material in this section. Dimakopoulou et al. (2021a)

Definition 15 (Propensity Score) *The propensity score of arm i at time t is the probability with which it is chosen, given the history $\mathbb{H}_{t-1}$ of the previously chosen arms and their outputted rewards:*

$$\pi_{t,i} = \mathbb{P}[a_t = i | \mathbb{H}_{t-1}]$$

Definition 16 (Inverse Propensity Score Weighting (IPW)) *Hadad et al. (2021) Assuming that the propensity scores are accurate, IPW provides an unbiased estimate of the true mean reward of an arm I:*

$$r_{t,i}^{IPW} = \frac{1}{t} \sum_{s=1}^{t} \frac{1[a_s = i]}{\pi_{s,i} r_s}$$

Definition 17 (Efficient Score) *The efficient score is an update to the previous estimate of the reward using the IPW and the newly acquired reward:*

$$\Gamma_{s,i} = r_{s_1,i} + \frac{1[a_s = i]}{\pi_{s,i}} (r_s - r_{s_1,i})$$

Definition 18 (Doubly-Robust Estimator (DR)) *Assuming that the propensity scores are accurate, DR shifts the previous estimate of the reward mean using the newly acquired reward and the IPW:*

$$r_{t,i}^{DR} = \frac{1}{t}\sum_{s=1}^{t}\Gamma_{s,i}$$

Definition 19 (Adaptive Doubly-Robust Estimator (ADR)) *Luedtke and van der Laan (2016) Rather than the DR's uniform weight of $\frac{1}{t}$, ADR uses a nonuniform weight $w_{s,i}$ which is adapted to $\mathbb{H}_{s-1}$.*

$$r_{t,i}^{ADR} = \frac{\sum_{s=1}^{t} w_{s,i}\Gamma_{s,i}}{\sum_{s=1}^{t} w_{s,i}}$$

For the purposes of this paper, we define $w_{s,i} = \sqrt{\pi_{s,i}}$.

3.1.2 Doubly-Adaptive Thompson Sampling Algorithm

Now we describe the doubly-adaptive Thompson Sampling Algorithm (DATS). Directly building off of Adaptive Doubly-Robust Estimators, DATS assumes that the sampling distribution of arm i at time t is chosen to be normal $\mathcal{N}(\hat{\mu}_{t,i}, \hat{\sigma}^2 t, i)$, such that:

$$\hat{\mu}_{t,i} = r_{t,i}^{ADR}$$
$$\hat{\sigma}_{t,i}^2 = \frac{\sum_{s=1}^{t} \pi_{s,i}[(\Gamma - \hat{\mu}_{t,i})^1]}{(\sum_{s=1}^{t}\sqrt{\pi_{s,i}})^2}$$

The probability of sampling arm i at time t is the same as the probability for which a sample reward drawn from arm i's distribution is larger than all other sample rewards drawn from all other arms (i.e. the probability that, if we were to pull all the arms, arm i has the optimal reward). To prevent diminishing propensity scores, we also remove any arms for which this probability is falls below a threshold $\frac{1}{T}$.

Algorithm 7 Doubly-Adaptive Thompson Sampling Algorithm

For each arm $k = 1, \ldots, K$, set $S_k = 0, F_k = 0$
for i=1,..., K **do**
    Play arm i and observe reward $r_{0,i}$
    Initialize $\hat{r}_{0,i} = r_{0,1}$, $\pi_{1,i} = \frac{1}{K}$
end for
for t=1,2,... **do**
    From distribution a_t $Multinomial(A_t, (\pi_{t,a})_{a\in A_t})$, draw an arm a_t from set of optimal arms A_t, play it, and observe reward r_t
    Calculate efficient score and update sampling distribution for arm t accordingly
    Remove arms that fall below a threshold value of draw probability
    Compute the new propensity scores
end for

3.1.3 Significance of DATS

Empirical experiments performed by Dimakopoulou et al. (2021b) show that DATS improved performance over A-B testing and the previous variants on Thompson Sampling in terms of both cumulative regret and sample complexity. This demonstrates that the adaptive weights schema used by DATS allows it to use the unbiased sample mean without suffering from uncontrolled variance, unlike previous approaches. Dimakopoulou et al. (2021b) also show theoretically that the expected regret for DATS is upper bounded by $O(\sqrt{K^2 T log(T)})$ for the K-arm bandit case.

More practically, for applications such as web-service testing or clinical trials where resources are limited and other approaches (such as A/B testing) may be cost-prohibitive, DATS demonstrates an

alternative where reliable inference can be produced even in the online setting. In improving both the reward at test time and the sample complexity needed to identify the optimal arm, DATS shows clear improvement over previous approaches and is a promising improvement on the standard Thompson sampling approach.

3.2 Variational Bayesian Optimistic Sampling

Variational Bayesian optimistic sampling (VBOS) is a Bayesian approach to online learning O'Donoghue and Lattimore (2021). At each step, the VBOS algorithm solves a convex optimization problem over the simplex. The solution to the problem will be a policy that satisfies a particular optimal condition. VBOS is similar to the Thompson Sampling algorithm mentioned before.

In O'Donoghue and Lattimore (2021), the authors first provide an upper bound for the conditional expectation. They denote by $\Psi_X : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ the cumulant generating function of $X - \mathbb{E}[X]$

$$\Psi_X(\beta) = \log \mathbb{E}[\exp(\beta^T(X - \mathbb{E}[X]))]$$

Then they present the following lemma for bounding the conditional expectation:

Lemma 20 *Let $X : \Omega \to \mathbb{R}$ be a random variable on $(\Omega, \mathcal{F}, \mathbb{P})$ satisfying $X \in L_1$, and let $A \in \mathcal{F}$ be an event with $\mathbb{P}(A) > 0$. Then for any $\tau > 0$,*

$$\mathbb{E}[X|A] \leq \mathbb{E}[X] + \tau\Psi_X(1/\tau) - \tau \log \mathbb{P}(A)$$

From lemma 20 we can deduce the following theorem:

Theorem 21 *Let $X : \Omega \to \mathbb{R}$ be a random variable such that the interior of the domain of Ψ_X is non-empty, then under the same assumptions as Lemma 18 we have*

$$\mathbb{E}[X|A] \leq \mathbb{E}[X] + (\Psi_X^*)^{-1}(-\log \mathbb{P}(A))$$

Then using Theorem 21 we can derive the maximal inequality to bound the regret

Lemma 22 *Let $\mu : \Omega \to \mathbb{R}^A, \mu \in L_1^A$ be a random variable, let $i^* = \arg\max_i \mu_i$ and denote by $\Psi_i := \Psi_{\mu_i}$, then*

$$\mathbb{E}[\max_i \mu_i] \leq \sum_{i=1}^{A} \mathbb{P}(i^* = i) \left(\mathbb{E}[\mu_i] + (\Psi_i^*)^{-1}(-\log \mathbb{P}(i^* = i))\right)$$

Then we need to introduce a 'optimism' map $\mathcal{G}_\phi^t : \Delta_A \to \mathbb{R}$ which for a random variable $\mu : \Omega \to \mathbb{R}^A$ distributed according to $\phi(\cdot|\mathbb{F}_t)$ is given by

$$\mathcal{G}_\phi^t(\pi) := \sum_{i=1}^{A} \pi_i \left(\mathbb{E}^t[\mu_i] + (\Psi_i^{t^*})^{-1}(-\log \pi_i)\right)$$

where Δ_A denotes the probability simplex of dimension $A - 1$ and $\Psi_i^t = \Psi_{\mu_i|\mathcal{F}_t}$

With the notation, we can now define the optimist set as the following:

Definition 23 (Optimistic set) *Let $\mu : \Omega \to \mathbb{R}^A$ be a random variable distributed according to $\phi(\cdot|\mathcal{F}_t)$, then the optimistic set is*

$$\mathcal{P}_\phi^t := \{\pi \in \Delta_A | \mathbb{E}^t[\max_i \mu_i] \leq \mathcal{G}_\phi^t(\pi)\}$$

Now we can define the variational Bayesian optimistic sampling algorithm. VBOS produces a policy $\pi^t \in \mathcal{P}_\phi^t$ at each round by construction. The maximum is guaranteed to exist since Δ_A is compact and $\mathcal{G}_\phi^t$ is continuous. We cam summarize the VBOS as follows:

Algorithm 8 VBOS for bandits

1: **for** t=1,2,..., T **do**
2: compute $\pi^t = \arg\max_{\pi \in \Delta_A} \mathcal{G}_\phi^t(\pi)$
3: sample $a_t \sim \pi^t$
4: **end for**

The paper O'Donoghue and Lattimore (2021) also gives an analysis of the regret for VBOS, which states that

Lemma 24 *Let ALG produce any sequence of policies π^t for $t = 1, \dots, T$ that satisfy $\pi^t \in \mathcal{P}^t_\phi$, then*

$$BayesRegret(\phi, ALG, T) \leq \mathbb{E} \sum_{t=1}^{T} \sum_{i=1}^{A} \pi_i^t (\Psi_i^{t^*})^{-1}(-\log \pi_i^t)$$

With the help of Lemma 24, we can prove the following theorem:

Theorem 25 *Let ALG produce any sequence of policies π^t for $t = 1, \dots, T$, that satisfy $\pi^t \in \mathcal{P}^t_\phi$ and assume that both the prior and reward noise are 1-sub-Gaussian for each arm, then the Bayes regret is*

$$\textit{BayesRegret}(\phi, \textit{ALG}, T) \leq \sqrt{2AT \log A(1 + \log T)} = \tilde{O}(\sqrt{AT})$$

Proof. Since the prior and noise term are 1-sub-Gaussian for each arm, we can bound the cumulant generating function of μ_i at time t as

$$\Psi_i^t(\beta) \leq \frac{\beta^2}{2(n_i^t + 1)}$$

where n_i^t is the number of observations of arm i before time t. Then we have the following bound for $y \geq 0$

$$(\Psi_i^{t^*})^{-1}(y) \leq \sqrt{\frac{2y}{n_i^t + 1}}$$

Combining this with Lemma 24, we have

$$\begin{aligned} BayesRegret(\phi, ALG, T) &\leq \mathbb{E} \sum_{t=1}^{T} \sum_{i=1}^{A} \pi_i^t (\Psi_i^{t^*})^{-1}(-\log \pi_i^t) \\ &\leq \mathbb{E} \sum_{t=1}^{T} \sum_{i=1}^{A} \pi_i^t \sqrt{\frac{-2 \log \pi_t}{n_i^t + 1}} \\ &\leq \mathbb{E} \sqrt{\sum_{t=1}^{T} H(\pi^t) \sum_{t=1}^{T} \sum_{i=1}^{A} \frac{2\pi_t}{n_i^t + 1}} \end{aligned}$$

which follows from the Cauchy-Schwarz inequality. To conclude the proof we use the fact that $H(\pi_t) \leq \log(A)$ and the pigeonhole principle.

■

3.3 Neural Architecture Search via Combinatorial Multi-armed Bandit

Finally, we would like to mention the work of Huang et al. (2021), a fascinating expansion of the multi-armed bandit formulation to address a problem which is relevant for practical machine learning. The authors approach the problem of Network Architecture Search (NAS), a complicated high-dimensional search problem which aims to find the parameters and structures of a successful deep neural network before it is trained. Since this search is conducted over a potentially infinite space, NAS approaches often use reinforcement learning formulations such as the multi-armed bandit.

In this work, the authors formulate the NAS problem as a Combinatorial Multi-Armed Bandit problem. In this modification of the original problem, a global multi-armed bandit problem is decomposed into many local small multi-armed bandit problems. At each iteration, a global multi-armed bandit algorithm picks which local problem to execute, and a local algorithm will make the decision for that smaller problem.

At the heart of this transformation is a *monoticity assumption*. This assumption is that the the global reward is approximated well by the sum of the local rewards. Concretely,

$$\mu_g \approx \sum_i \mu_i$$

where μ_g is the mean global reward and each μ_i is the mean reward for a particular local subproblem. The intuition behind this assumption is that if each local subproblem is relatively independent, then each one can be optimized separately, cutting down the overall size of the search space. This approach clearly is successful in network architecture search, in which different architecture parameters apparently operate independently of each other.

This tiered structure allows the overall algorithm to explore the high dimensional search space much more efficiently than a standard bandit algorithm. In their evaluation section, the authors show that this combinatorial approach achieves results that are comparable to the state of the art in 1/20 of the overall training time. This unique way of modifying the multi-armed bandit formulation leads to a powerful benefit for end users seeking to explore neural network architectures.

4 Conclusion

In this report, we began by introducing the setup of the problem and the main way multi-armed bandit algorithms are evaluated: regret. We continued by describing and mathematically analyzing three classic algorithmic approaches to the problem: epsilon-greedy, upper confidence bound, and thompson sampling. Finally, we explored three modern additions to the literature - an expansion upon thompson sampling, an expansion based on bayesian samping, and a combinatorial decomposition into local and global bandits. The multi-armed bandit problem is a foundational and important formulation in reinforcement learning and online learning algorithms, and continues to be a rich subfield for future work.

References

Shipra Agrawal. 2019. Lecture 3: Ucb algorithm.

Shipra Agrawal and Navin Goyal. 2011. Analysis of thompson sampling for the multi-armed bandit problem. *CoRR*, abs/1111.1797.

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

Nicolò Cesa-Bianchi and Paul Fischer. 1998. Finite-time regret bounds for the multiarmed bandit problem. In *In 5th International Conference on Machine Learning*, pages 100–108. Morgan Kaufmann.

Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Maria Dimakopoulou, Zhimei Ren, and Zhengyuan Zhou. 2021a. Doubly-adaptive thompson sampling for multi-armed and contextual bandits. *CoRR*, abs/2102.13202.

Maria Dimakopoulou, Zhimei Ren, and Zhengyuan Zhou. 2021b. Online multi-armed bandits with adaptive inference.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(39):1079–1105.

Vitor Hadad, David A. Hirshberg, Ruohan Zhan, Stefan Wager, and Susan Athey. 2021. Confidence intervals for policy evaluation in adaptive experiments.

Hanxun Huang, Xingjun Ma, Sarah M. Erfani, and James Bailey. 2021. Neural architecture search via combinatorial multi-armed bandit. *CoRR*, abs/2101.00336.

T.L Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22.

Alexander R. Luedtke and Mark J. van der Laan. 2016. Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy. *The Annals of Statistics*, 44(2):713 – 742.

Brendan O'Donoghue and Tor Lattimore. 2021. Variational bayesian optimistic sampling.

Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527 – 535.

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, and Ian Osband. 2017. A tutorial on thompson sampling. *CoRR*, abs/1707.02038.

Aleksandrs Slivkins. 2021. Introduction to multi-armed bandits.

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

Christopher Watkins. 1989. Learning from delayed rewards.