A BAYESIAN FRAMEWORK FOR EXPLORATION IN REINFORCEMENT LEARNING

JOHN-WESLEY APPLETON [JOHNWES@SEAS]

ABSTRACT. In this paper, we investigate the use of Thompson Sampling, a Bayesian sampling technique, for solving Multi-actioned Bandit (MAB) problems, a simplified model for reinforment learning (RL) problems. We compare Thompson Sampling to the widely used Epsilon-Greedy algorithm on a simulated MAB problem. Our results show that Thompson Sampling outperforms Epsilon-Greedy in terms of cumulative reward, especially when the rewards of the actions are highly uncertain. Moreover, our findings demonstrate that the Thompson Sampling algorithm selects the optimal action more frequently than the Epsilon-Greedy algorithm. Our contributions include a thorough explanation of MAB problems and the Epsilon-Greedy and Thompson Sampling algorithms, as well as a detailed comparison of the two algorithms on a simulated MAB problem. Our findings suggest that Thompson Sampling is a promising approach for designing optimal behavior in robotics applications.

1. INTRODUCTION

Robotics has emerged as a crucial field with applications ranging from manufacturing and transportation to healthcare. With robots becoming more prevalent in our daily lives, it is imperative that they function optimally to avoid serious injuries and costly repairs. One of the critical challenges in robotics is developing algorithms that enable robots to learn and adapt to new environments and tasks. Reinforcement learning (RL) has become a popular technique in robotics for this purpose. However, RL algorithms require exploration, which can be expensive and prevent the robot from acting optimally.

Recent advancements in RL have focused on designing algorithms that balance exploration and exploitation to achieve optimal performance. Multi-Armed Bandit (MAB) problems are a popular class of RL problems that involve a single agent interacting with multiple actions (or ”arms”) and attempting to maximize its cumulative reward over time. One approach to solving MAB problems is the Epsilon-Greedy algorithm, which is simple to implement but may not always be optimal. Despite recent advancements, there are still some problems with current RL algorithms that prevent their widespread adoption in robotics. One of these problems is the requirement for exploration, which can be expensive in real-world scenarios.

1.1. Contributions.

In this paper, we explore one solution to this problem by investigating the performance of a Bayesian sampling technique called Thompson sampling in MAB problems. Specifically, we compare Thompson sampling to the Epsilon-Greedy approach and evaluate its potential for improving RL algorithms in robotics. By addressing this critical issue, we hope to contribute to the development of more efficient and effective RL algorithms that can be implemented on robots to improve their performance and safety.

2. BACKGROUND

MAB problems are a popular class of RL problems. In MAB problems, an agent interacts with a set of actions and receives a reward for each interaction. The goal of the agent is to determine which action to take to maximize its total reward over time. MAB problems are often used as a simplified model for more complex RL problems, such as robot control and game playing.

The Epsilon-Greedy algorithm is a simple and widely used approach to solving MAB problems. In the Epsilon-Greedy algorithm, the agent selects the action with the highest estimated reward with probability $1 - \epsilon$, and selects a random action with probability ϵ. The value of ϵ is a parameter that determines the degree of exploration versus exploitation. The Epsilon-Greedy algorithm is easy to implement and can be effective in certain scenarios. However, it may not always be optimal, particularly in scenarios where the rewards of the action are highly uncertain.

Thompson Sampling is a Bayesian sampling technique that has gained popularity in MAB problems. In Thompson Sampling, the agent maintains a probability distribution over the true reward of each action. The probability distribution represents the agent's belief about the true reward of each action, and is updated based on the observed rewards. At each interaction, the agent selects an action according to the probability distribution. The action with the highest probability of having the highest reward is selected. Thompson Sampling has been shown to outperform Epsilon-Greedy in certain scenarios, particularly when the rewards of the actions are highly uncertain.

3. Related Work

MAB problems have been widely studied in the literature due to their simplicity and relevance to practical problems. An influention work on MAB problems is ”Introduction to Multi-Armed Bandits” by Aleksandrs Slivkins[3], where the author provides a comprehensive introduction to MAB problems, their applications, and existing solution approaches.

Another related study is ”Multi-Armed Bandits in Recommendation Systems: A survey of the state-of-the-art and future directions” by Silva et al.[1], which surveys various MAB algorithms that have been applied to recommendation systems. The study provides a comprehensive overview of MAB algorithms and their applications in real-world problems.

Several studies have compared the performance of different algorithms for solving MAB problems. Anupam Singh's ”Reinforcement Learning Based Empirical Comparison of UCB, Epsilon-Greedy, and Thompson Sampling”[2] provides an empirical comparison of three popular MAB algorithms, including Epsilon-Greedy and Thompson Sampling.

Our study aims to extend prior research by investigating the effectiveness of Epsilon-Greedy and Thompson Sampling algorithms on a simulated MAB problem. We replicate the experiment conducted by Anupam Singh to examine whether their findings are consistent in a different environment. Furthermore, we provide the code for our implementation, making it available for future use by researchers.

4. Approach

In this section, we describe our approach for comparing the performance of the Epsilon-Greedy and Thompson Sampling algorithms on a simulated MAB problem. We first provide an overview of the simulated MAB environment, followed by a description of the implementation of the two algorithms.

4.1. **Multi Armed Bandit Environment.**

In our simulated MAB environment, we consider a set of K actions, each with an unknown reward distribution. Let $r_{i,t}$ denote the reward obtained from action i at time t. We assume that the rewards are independent and identically distributed (i.i.d.) across time steps and actions. The goal of the agent is to maximize the cumulative reward obtained over a fixed time horizon T:

$$\max_{a_1,\ldots,a_T} \sum_{t=1}^{T} r_{a_t,t}$$

Where a_t is the action selected by the agent at time t. We assume that the agent can observe the reward obtained at each time step, but has no prior knowledge about the reward distributions of the action. In our simulated environment, we generate the reward distributions for each action from a Bernoulli distribution:

$$r_{i,\cdot} \sim \text{Bernoulli}(\theta_i)$$

Where we set θ_k to a random value drawn from the interval [0,1]. The agent's goal is to learn the optimal action, which is the action with the highest mean reward: $\mu^* = \max_i \theta_i$.

4.2. **Epsilon-Greedy Algorithm.**

The Epsilon-Greedy algorithm is a simple and widely-used algorithm for solving the MAB problem. At each time step t, the agent chooses the action with the highest empirical mean reward $\hat{\mu}_i(t)$ with probability $1 - \epsilon_t$, and chooses a random action with probability ϵ_t. The parameter ϵ_t controls the exploration-exploitation trade-off: when ϵ_t is high, the agent is more likely to choose a random action and explore the reward distributions of the action, while when ϵ_t is low, the agent is more likely to choose the action with the highest empirical mean reward and exploit the knowledge it has gained so far.

Formally, let $N_i(t)$ denote the number of times action i has been selected up to time t, and let $X_{i,t}$ denote the reward obtained from action i at time t. The empirical mean reward of action i up to time t is given by:

$$\hat{\mu}_i(t) = \frac{1}{N_i(t)} \sum_{s=1}^{t} X_{i,s}$$

At each time step t, the agent selects an action a_t according to the following rule:

$$a_t = \begin{cases} \arg\max_i \hat{\mu}_i(t) & \text{with probability} \quad 1 - \epsilon \\ \text{a randomly chosen action} & \text{with probability} \quad \epsilon \end{cases}$$

4.3. **Thompson Sampling.** Thompson Sampling is a Bayesian algorithm that involves updating a prior distribution with each observation to obtain a posterior distribution. The posterior distribution is then used to generate a probability distribution over the expected rewards of each action, which is then used to select the next action to play. In our implementation, we use a conjugate prior of the Bernoulli distribution, which is the Beta distribution. We initialize the Beta distribution for each action with parameters $\alpha = 1$ and $\beta = 1$, which corresponds to a uniform prior. At each time step t, after playing action i and receiving reward $X_i(t)$, we update the posterior distribution for action i using the following formula:

$$p_i \mid \alpha, \beta, X, t \sim \text{Beta}\left(\alpha + \sum_{s=1}^{t} X_i(s), \beta + \sum_{s=1}^{t} 1 - X_i(s)\right)$$

At every time step t, we obtain posterior samples p_i from the posterior distribution of each action i. The agent selects an action a_t according to the following rule:

$$a_t = \arg\max_i p_i$$

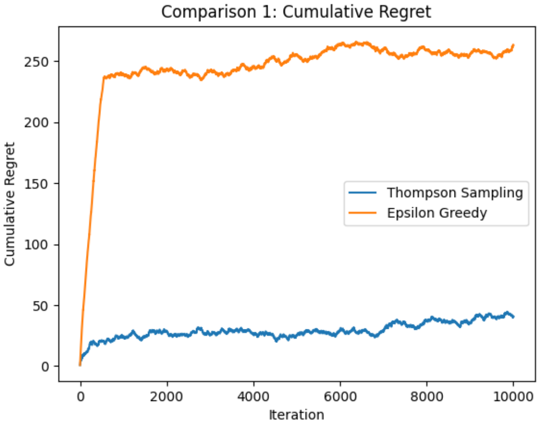
Our implementation approach involved developing Python code for both algorithms and conducting experiments to compare their performance. We present our experimental setup and results in the next section.

5. Experimental Results

In this section, we present the empirical results of our implementation of the Epsilon-Greedy and Thompson Sampling algorithms on the MAB environment. We compare the performance of both algorithms using three metrics: cumulative regret, percent of optimal actions, and average reward. These metrics will provide insight into the effectiveness of each algorithm in selecting the optimal action over time.

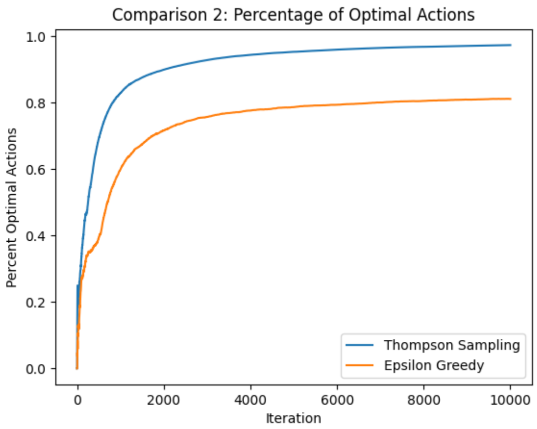
5.1. **Cumulative Regret.** Cumulative regret is a metric used to evaluate the performance of an algorithm in selecting the optimal action over time. It is defined as the difference between the cumulative reward obtained by the algorithm and the cumulative reward that would have been obtained if the optimal action had been selected at each time step. In other words, it measures the regret of not selecting the optimal action. A lower cumulative regret indicates a better algorithm performance.

In our experiment, we found that Thompson sampling had a much lower cumulative regret than Epsilon-Greedy. This indicates that Thompson sampling was better at selecting the optimal action over time. Specifically, after 1000 iterations, the cumulative regret of Thompson sampling was around 30 while that of Epsilon-Greedy was around 250. This suggests that Thompson sampling is a more effective algorithm for the MAB problem, as it is able to minimize the regret of not selecting the optimal action.



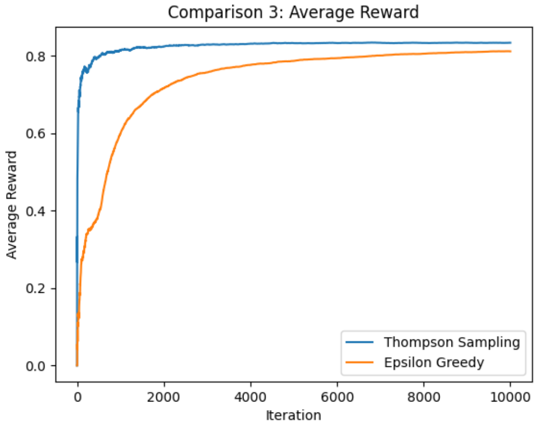
5.2. **Percent of Optimal Actions.** Percent of optimal actions is another metric that is used to evaluate the performance of MAB algorithms. It measures the proportion of times the algorithm chooses the optimal action among all the selections made. This metric is calculated as the number of times the optimal action is chosen divided by the total number of selections. It is a good metric because it measures the ability of the algorithm to learn which action is the best and choose it more often.

In our experiment, we found that Thompson sampling converges to a higher percentage of optimal actions compared to Epsilon-Greedy. As the number of rounds increases, the percentage of optimal actions chosen by Thompson sampling approaches the theoretical maximum, while the percentage chosen by Epsilon-Greedy plateaus at a lower value. This suggests that Thompson sampling is better able to learn which action is the best and exploit that information to choose the optimal action more often.



5.3. **Average Reward.** Average reward is another commonly used evaluation metric for MAB algorithms, and it refers to the average amount of reward received by the algorithm over time. It is calculated by dividing the total reward accumulated by the number of time steps. This metric is useful because it measures the overall effectiveness of the algorithm in maximizing reward.

In our experiment, we found that Thompson sampling had a higher average reward compared to Epsilon-Greedy. This is consistent with our previous results, as the higher percentage of optimal actions chosen by Thompson sampling led to a greater amount of reward received. These findings support the effectiveness of Thompson sampling over Epsilon-Greedy in MAB problems.



6. Discussion

In this study, we compared the performance of the Epsilon-Greedy and Thompson Sampling algorithms on a simulated MAB problem. Our results show that Thompson Sampling outperformed Epsilon-Greedy in terms of cumulative regret, percent of optimal actions, and average reward. These findings are consistent with previous research that has also shown Thompson Sampling to be a more effective algorithm for MAB problems.

However, there are still limitations to this study. One of the limitations is that we only considered a single MAB problem with a fixed number of action and distribution parameters. In the future, it would be interesting to test the algorithms on a wider range of MAB problems with varying complexities. Additionally, we used a simulated environment instead of a real-world robotic system. Future studies could explore the use of these algorithms in actual robotic systems to determine their effectiveness in real-world scenarios.

In the broader context of reinforcement learning and robotics, our findings highlight the importance of exploration in RL algorithms. While exploration is necessary to achieve optimal performance, it can be expensive and potentially dangerous in robotic systems. Therefore, the development of more efficient exploration strategies, such as Thompson Sampling, can have significant implications for the safety and effectiveness of robotic systems. Overall, this study contributes to the ongoing effort to design better RL algorithms and improve the performance of robotic systems.

REFERENCES

[1] Nícollas Silva, Heitor Werneck, Thiago Silva, Adriano C.M. Pereira, and Leonardo Rocha. Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications*, 197:116669, 2022.

[2] Anupam Singh. Reinforcement learning based empirical comparison of ucb, epsilon-greedy, and thompson sampling. *Int. J. of Aquatic Science*, 12(2):2961–2969, 2021.

[3] Aleksandrs Slivkins. Introduction to multi-armed bandits. *CoRR*, abs/1904.07272, 2019.