Optimizing Movie Recommendation Systems with Multi-Armed Bandit Algorithms

Yaosheng Jian *

School of Intelligent Systems Science and Engineering/JNU-Industry School of Artificial Intelligence, Jinan University, Zhuhai, 519070, China

* Corresponding Author Email: jianyaosheng@stu2021.jnu.edu.cn

Abstract. In the current digital era, with the proliferation of online content, users are often overwhelmed by an abundance of digital advertisements, a consequence of the lower costs associated with digital channels compared to traditional media. Amidst this information deluge, recommendation systems have become pivotal in offering tailored suggestions to users. The Multi-Armed Bandit (MAB) problem, rooted in reinforcement learning, presents an effective approach for addressing the exploration-exploitation trade-offs inherent in recommendation systems. This paper specifically examines the implementation of MAB algorithms within the context of movie recommendation systems. A thorough literature review is conducted, focusing on the synergy between reinforcement learning and MAB algorithms, their role in recommendation systems, and an overview of fundamental MAB algorithms. To assess the efficacy of these algorithms, an experimental approach is designed, leveraging real-world datasets. This approach encompasses the formulation of the problem, data collection methods, selection of hyperparameters, and analysis of empirical results. The findings from these experiments indicate that Thompson Sampling stands out as the most effective MAB algorithm in the context of the studied datasets. The results of this study suggest that MAB algorithms have considerable potential to enhance the effectiveness of movie recommendation systems. Furthermore, the methodology and framework proposed in this paper offer valuable insights for integrating and combining multiple algorithms in complex decision-making scenarios.

Keywords: Multi-armed bandit algorithms; Recommendation system; Reinforcement learning.

1. Introduction

Reinforcement Learning (RL) has emerged as a dynamic and influential research area within machine learning, focusing on empowering agents to learn optimal decision-making strategies through environmental interactions. A core challenge within RL is the Multi-Armed Bandit problem, which necessitates sequential decision-making in the face of uncertain rewards.

The application of RL and MAB algorithms has garnered significant interest across various fields, including recommendation systems, robotics, finance, healthcare, and others [1]. This diverse applicability has spurred the development of innovative algorithms and techniques tailored to the distinct challenges encountered in these domains. In recommendation systems, for example, the decision-making process involves selecting items for user recommendations, aiming to maximize engagement and retention. This paper endeavors to evaluate the efficacy of different MAB algorithms within the context of recommendation systems. The author commences with a comprehensive review of relevant literature on MAB, particularly its integration into recommendation systems, and discusses critical metrics for appraising MAB algorithm performance. Following this, the paper outlines foundational MAB algorithms, such as ε-greedy, Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), and Thompson Sampling.

In the empirical section, the author meticulously details the experimental methodology, including problem definition, experimental framework, data sources, and hyperparameter selection. The subsequent segments present empirical results and a comparative analysis of each algorithm's performance, offering insightful perspectives on the practical efficacy of MAB algorithms in recommendation system contexts.

2. Literature Review

2.1. Intersection of Reinforcement Learning and Multi-Armed Bandits

Reinforcement Learning is a machine learning method aimed at learning optimal behavior through interactions with the environment. It is typically used for decision-making tasks, such as control, games, robotics, and more. In reinforcement learning, algorithms try different actions and learn which ones yield better rewards based on feedback from the environment [2].

In 1933, William R. Thompson introduced bandit problems in an article published in Biometrika. Thompson's focus was on medical trials and the ethical concerns of conducting blind trials without adjusting treatment allocations in real-time based on the efficacy of the drug [3]. The multi-armed bandit problem is a classic decision-making problem extended from bandit problems. In this problem, there is a slot machine with multiple arms to choose from. Each arm corresponds to a different probability distribution of rewards, and the goal is for the player to maximize their cumulative reward by trying different arms. The challenge lies in the fact that the player does not initially know the reward probabilities associated with each arm. Therefore, the player needs to balance exploration (trying new arms) and exploitation (choosing arms that are known to be better) to find the optimal arm selection strategy. This involves a trade-off between exploring new arms to gather more information and exploiting arms that are already known to provide higher rewards [4].

Multi-armed bandits and reinforcement learning have a common intersection lies in their shared focus on decision-making under uncertainty and optimizing long-term rewards [5]. While they have distinct characteristics, there are areas where MAB and RL overlap and can be applied together.

MAB algorithms, such as Thompson Sampling or Upper Confidence Bound, are often used as exploration-exploitation strategies within RL frameworks. In RL, agents learn through trial and error by interacting with an environment, similar to the exploration phase in MAB. MAB algorithms can help balance exploration and exploitation, allowing RL agents to efficiently seek out optimal actions while learning.

Additionally, MAB algorithms can be used as components of RL algorithms. For example, in contextual bandits, where each arm has a context or state associated with it, MAB techniques can be employed to determine which action to take given the current context. This context-dependent decision-making is a key component of RL.

The intersection of MAB and RL allows for the integration of exploration-exploitation strategies, context-dependent decision-making, and so on, enhancing the performance and adaptability of reinforcement learning algorithms in uncertain environments.

2.2. Application of the MAB Problem in Recommendation Systems

With the development of the Internet, users are bombarded with a vast number of online advertisements. Due to the relatively low cost of Internet advertising compared to other channels, companies flood the digital sphere with thousands of promotional campaigns [6]. From the perspective of media providers, it is crucial to showcase these advertisements to capture users' attention and maximize revenue generation. At this point, the Multi-Armed Bandit problem is crucial for recommendation systems which involves using MAB algorithms to optimize the selection and recommendation of items to users in real-time.

In a recommendation system, the MAB problem can be framed as the challenge of selecting the most relevant items (such as products, articles, or media content) to recommend to users, while balancing the trade-off between exploring new items to learn more about user preferences and exploiting known high-quality items to maximize user engagement [7]. The problems solved by the MAB algorithm in recommendation systems include: Exploration-Exploitation Trade-off, Personalization, and Real-time Adaptation.

MAB algorithms enable recommendation systems to balance exploration and exploitation. By exploring different items, the system gathers information about user preferences and item performance. This information is then used to exploit the best-performing items to maximize user

satisfaction, which becomes an important reason for achieving personalization. Subsequently, MAB-based recommendation systems dynamically adjust item selection strategies based on real-time feedback. As user preferences and item performance change, the system can adapt its recommendations accordingly [8].

2.3. Overview of Prominent MAB Algorithms

2.3.1. The Regret

In Multi-Armed Bandits problems, regret is defined by "Reward lost by taking sub-optimal decisions", which is used to measure the performance of a strategy. In Multi-Armed Bandits problems, the smaller the regret value, the better the performance of the strategy. Therefore, the ultimate goal of the Multi-Armed Bandits problems is to minimize regret and achieve the best rewards. In the textbook Bandit Algorithms, the authors use the following formula (1) to define regret [9]:

$$R_n(\pi, \quad \nu) = n\mu^*(\nu) - E[\sum_{t=1}^{n} X_t] \tag{1}$$

Where $n\mu^*(\nu)$ is the max expected reward if the best arm is known and $E[\sum_{t=1}^{n} X_t]$ is the expected cumulative reward of the policy π.

2.3.2. ε-greedy Algorithm

When dealing with Multi-Armed Bandits problems or reinforcement learning tasks, balancing exploration and exploitation is the key to the problem. The ε-greedy algorithm is a strategy that finds a balance between exploration and exploitation.

In ε-greedy algorithm, there is a hyperparameter ε, which defines the probability beyond adopting an exploitation strategy (i.e. selecting the currently best option). Specifically, there is a ε chance of exploring a new option (i.e. not on the current optimal choice) with a 1-ε chance of using current knowledge (i.e. selecting the currently optimal option) [10].

2.3.3. Explore-Then-Commit Algorithm

ETC explores by engaging in a fixed number of interactions for each arm, and then exploits by committing to the arm that emerged as the best during the exploration phase. ETC utilizes a hyperparameter m to indicate the number of times it explores each arm. As there are k arms available, ETC conducts m×k rounds of exploration before choosing the optimal arm for the remaining phase. Intuitively speaking, the exploration phase expands as m increases.

2.3.4. Upper Confidence Bound Algorithm and Asymptotically Optimal UCB Algorithm

The UCB algorithm operates on the principle of maintaining an optimistic outlook in uncertain situations, advocating for actions to be taken under the assumption that the environment is as favorable as reasonably feasible. For the UCB algorithm, the index of each arm i is defined as:

$$UCB_i(t-1) = \hat{\mu}_i(t-1) + \frac{B}{2}\sqrt{\frac{\tau \log n}{T_i(t-1)}} \tag{2}$$

Where B is the difference between the maximum possible reward value and the minimum possible reward value, $\hat{\mu}_i(t-1)$ is the mean reward of the i^{th} arm , $T_i(t-1)$ is the number of the selection of the i^{th} arm at the time t and the hyperparameter τ determines the size of upper confidence bound. For Asymptotically Optimal UCB, the index of each arm i adjusts to:

$$UCB_i(t-1) = \hat{\mu}_i(t-1) + \frac{B}{2}\sqrt{\frac{\tau \log(f(t))}{T_i(t-1)}} \tag{3}$$

Where $f(t) = 1 + t(\log(t))^2$, and other parameters can be referred to formula (2).

2.3.5. Thompson Sampling Algorithm

Thompson sampling was originally introduced by Thompson in 1933, which employs a Bayesian statistical framework to formulate $\hat{\mu}_i$ generated by the prior distribution. The Beta distribution is used in this text and the equation is as follow:

$$f_{prior}(x \mid \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{4}$$

Where $B(\alpha, \beta)$ a beta function and the posterior distribution of is $\hat{\mu}_i$ is as follow:

$$\begin{cases} \alpha_i = \alpha_i + 1, \beta_i = \beta_i, if(\hat{\mu}_i = 1) \\ \alpha_i = \alpha_i, \beta_i = \beta_i + 1, if(\hat{\mu}_i = 0) \end{cases} \tag{5}$$
$$f_{posterior}(\hat{u}_i \mid reward_i) = B(\alpha_i, \beta_i)$$

Where $reward_i$ is the observed value obtained?

3. Experimental Methodology

3.1. Formulation of the Problem

Movie recommendation systems can be considered as a subset or specialization within the broader field of advertising recommendation systems. The correlation between movie recommendation systems and advertising recommendation systems lies in their shared aim of providing customized content to users. Both systems employ user data to deliver personalized recommendations, with the ultimate goal of enhancing user experience and fostering user engagement. The objective of a movie recommendation system is to Recommend more precise movie genres to achieve higher click through rates for users by considering their historical rating data, preferences, and viewing history.

In this empirical study, movie ratings were continuously obtained using MAB algorithms to simulate a movie recommendation system and furthermore, each MAB algorithm will be evaluated and compared.

3.2. Experimental Framework

In this empirical study, real movie rating data was utilized, obtained from millions of user ratings. The experiment initially focuses on the configuration of algorithm hyperparameters to identify the optimal performance for each algorithm. Subsequently, a direct comparison of the actual performance of five MAB algorithms will be conducted. In addition, Python environment is used to collect dataset, edit the code and plot the graphics.

3.3. Data Sources and Collection

The dataset used in this empirical study is from “Movie Lens Dataset”, which contains movie rating information from millions of users. In the context of this study, movies are rated on a scale of 1 to 5, with a score of 3 indicating neutrality, a score below 3 indicating disinterest from the user, and a score above 3 indicating the user will click on the option (i.e., the user is more likely to click on the movie with a score above 3).

3.4. Hyperparameters Selection

3.4.1. ε-greedy Algorithm

ε-greedy algorithm has a hyperparameter, denoted as ε, which represents the probability of selecting a random choice. The algorithm is executed by varying the ϵ parameter from 0.1 to 0.5, incrementing it by 0.1 at each step. Intuitively speaking, as ε increases, the intensity of exploration also increases. However, with the increase of cost associated with exploration, algorithm will no

longer be efficient. Therefore, the author use ε=0.1 to participate in comparison of algorithms. As shown in Fig 1.

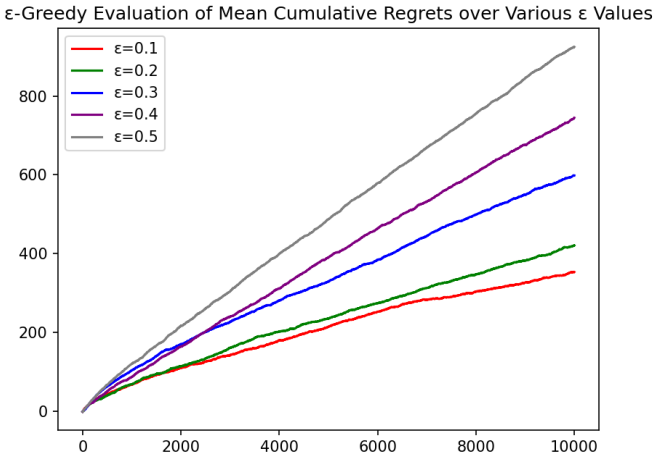

Fig 1. The figure shows the evaluation of mean cumulative regrets over various ε values (Photo/Picture credit: Original).

3.4.2. ETC Algorithm

ETC algorithm has a hyperparameter, denoted as m, which determines the length of the exploration phase. The setting of parameter m is a worth topic to discuss: a too small value of m will result in a too small exploration phase, making it difficult to get sufficient feedback for the algorithm to obtain the optimal choice. The algorithm is likely to choose a suboptimal option with an insufficient phase, and once suboptimal choice is made during the exploitation phase, the cumulative regret will increase linearly. Meanwhile, a too large value of m will result in an overly large exploration phase, causing the cumulative regret to increase excessively during the exploration phase, which means the cost of exploration is too high.

Base on the above discussion, the author attempts to set m×k as 10%~50% of N and the empirical study indicates that with the level of 10%,the algorithm makes a balance between exploration and exploitation(i.e. the algorithm is able to observe the optimal choice through sufficient feedback without incurring excessive exploration costs). As shown in Fig 2.

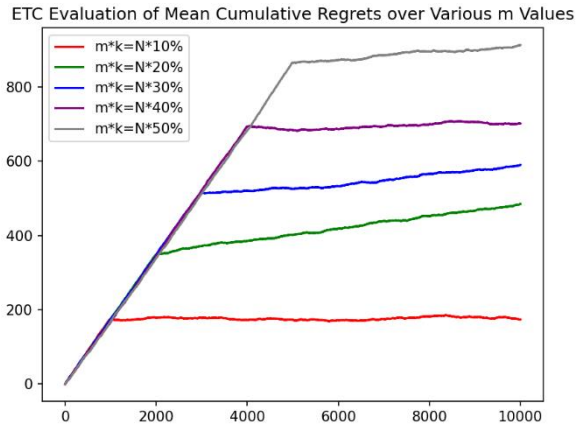

Fig 2. The figure shows the evaluation of mean cumulative regrets over various m values (Photo/Picture credit: Original).

3.4.3. UCB Algorithm

UCB algorithm has a hyperparameter, denoted as τ , which determines the size of upper confidence bound. The author attempts to set τ as 1 to 5, incrementing it by 1 at each step and the empirical study indicates that with τ =1, the algorithm performs most efficiently. As shown in Fig 3.

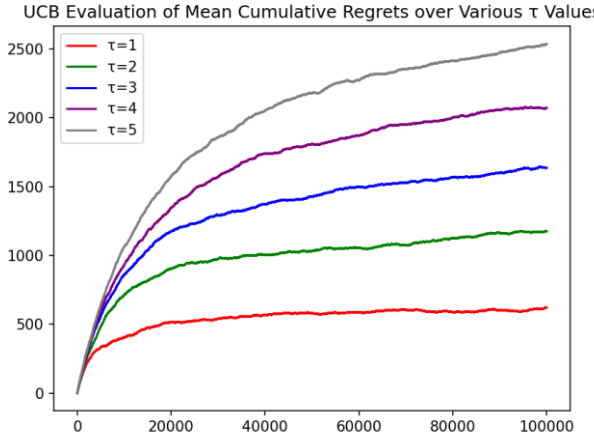

Fig 3. The figure shows the evaluation of mean cumulative regrets over various τ values (Photo/Picture credit: Original).

3.5. Empirical Results

ε-greedy, ETC and UCB have hyperparameters set, while Asymptotically Optimal UCB do not have hyperparameters. For ε-greedy, the author set ε as 0.1, which performs the best in the hyperparemeters set. Similarly, the author will set m×k as 10% and τ as 1 because they perform best in their respective algorithms. As shown in Fig 4.

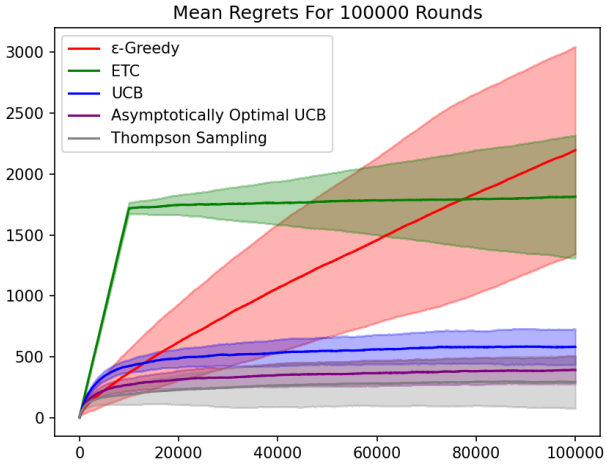

Fig 4. The figure shows the empirical results over various MAB algorithms (Photo/Picture credit: Original).

4. Comparative Analysis

The performance of five algorithms on the dataset is presented in the same graph. The results show that the performance of UCB, Asymptotically Optimal UCB and Thompson Sampling is significantly better than ε-greedy and ETC. For ε-greedy, due to its random exploration with a certain probability, it is intuitively to explain why its regret will show a linear growth phenomenon. Similarly, for ETC algorithm, it always requires a certain exploration rate. Therefore, in the case of a large sample size, its exploration phase will be infinitely magnified, resulting in linear growth of its cumulative regret during the exploration phase. As a result, compared to other algorithms, it does not have an advantage in large-scale datasets. UCB, Asymptotically Optimal UCB and Thompson Sampling all perform well, and their performance is stable. However, Algorithm A performs the best, indicating that in a recommendation system, Thompson Sampling is most suitable for adjusting decisions based on continuous feedback and making optimal choices.

5. Conclusions

In this empirical study, a substantial dataset of real movie ratings serves as a robust simulation environment, validating the practical applicability of various algorithms. The focus is on fine-tuning the hyperparameters of ε-greedy, ETC, and UCB algorithms to optimize their performance within this dataset. Through direct algorithmic comparisons, the study concludes that UCB, Asymptotic Optimistic UCB, and Thompson Sampling exhibit superior performance over ε-greedy and ETC. Notably, Thompson Sampling shows exceptional results, affirming its effectiveness in optimal decision-making and adaptability based on ongoing feedback within recommendation systems. These findings carry profound implications for the field of multi-armed bandit algorithms, indicating their potential for broader application, particularly in reinforcement learning contexts. By leveraging the balance between exploration and exploitation, these algorithms demonstrate significant promise for improving decision-making in dynamic environments such as online advertising, clinical trials, and resource allocation.

It is important to note that this research concentrates on the performance of individual algorithms against real-world datasets, without delving into the integration and synergy of multiple algorithms in more complex scenarios. The fusion and orchestration of various algorithms could further enhance their efficacy and adaptability to intricate situations, which is vital for their practical deployment. Additionally, exploring the impact of diverse problem contexts, such as non-stationary environments or scenarios with partial feedback, would yield valuable insights into the resilience and versatility of these algorithms. The implications of this empirical study on multi-armed bandit algorithms extend well beyond their immediate applications, suggesting broader potential in reinforcement learning scenarios. Continued exploration and research in this domain are poised to significantly advance decision-making methodologies across various sectors and contribute to the evolution of intelligent systems.

References

[1] Mambou, E. N., & Woungang, I. (2023). Bandit Algorithms Applied in Online Advertisement to Evaluate Click-Through Rates. In 2023 IEEE AFRICON (pp. 1-5). IEEE.

[2] Lattimore, T., & Szepesvári, C. (2020). Bandit Algorithms. Cambridge University Press.

[3] Slivkins, A. (2022). Introduction to Multi-Armed Bandits.

[4] Sondik, E. J. (1978). The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. Operations Research, 26(2), 282-304.

[5] Nishimura, Y. Ad Recommender System Analysis by the Multi-Armed Bandit Problem.

[6] Zhou, Q., Zhang, X., Xu, J., & Liang, B. (2017). Large-scale bandit approaches for recommender systems. In International Conference on Neural Information Processing (pp. 811-821). Springer.

[7] Bouneffouf, D., Laroche, R., Urvoy, T., Féraud, R., & Allesiardo, R. (2014). Contextual bandit for active learning: Active Thompson sampling. In International Conference on Neural Information Processing (pp. 405-412). Springer.

[8] Muqattash, I., & Hu, J. (2023). A ϵ-Greedy Multiarmed Bandit Approach to Markov Decision Processes. Stats, 6, 99-112.

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

[10] Elena, G., Milos, K., & Eugene, I. (2021). Survey of multiarmed bandit algorithms applied to recommendation systems. International Journal of Open Information Technologies, 9(4), 12-27.