Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/34/20230322

Investigation of frontier Multi-Armed Bandit algorithms and applications

Liangxu Wang

Department of Computer Science, Nankai University, Tianjin, 300350, China

2113198@mail.nankai.edu.cn

Abstract. Since the amount of content online is growing exponentially and people's time is limited, there is an urgent need for a high-performance algorithm that can make effective recommendations. This paper will introduce a recommendation system model, a sequential decision model, which is called Multi-Armed Bandit. The main idea of the Multi-Armed Bandit model is that at the beginning of the algorithm, all the recommended items are set to the same weight. In the subsequent recommendation process, the model explores the distribution of each item while changing the weight of each item according to the average revenue of each item, and selects more items with larger weight. This paper will introduce three cutting-edge Multi-Armed Bandit algorithms, their algorithmic ideas and their respective characteristics. The idea of Explore-Then-Commit (ETC) algorithm is to explore each item a certain number of times, and then select the best item for subsequent recommendation. The idea of the Upper Confidence Bound (UCB) algorithm is to represent the "exploration" and "exploitation" of each item by numerical values and add them to the UCB value, and select the item with the largest UCB value each time. The idea of TS is to first assume the distribution of each item, and then change the parameters of the distribution of each item based on the reward. At the end, this paper will introduce several scenarios where Multi-Armed Bandit algorithms can be used to give the reader an idea of how to use Multi-Armed Bandit algorithms.

Keywords: multi-armed bandit, reinforcement learning, recommendation system.

1. Introduction

In the contemporary era characterized by the proliferation of big data, content on various e-commerce platforms and video recommendation platforms is growing exponentially. In the face of complicated online information, it is impractical for users to find their favourite content on their own. Therefore, the development of a suitable recommendation system becomes a crucial requirement. The efficacy of such a system relies on achieving a considerable level of reward with respect to a specific metric, while simultaneously adhering to cost-effectiveness principles. Additionally, it cannot be too costly - it is impossible to train a model for every user and keep tweaking that model to predict their favourite items or videos. With the constraints of this condition, the Multi-Armed Bandit (MAB) algorithms from the Reinforcement Learning (RL) domain.

Multi-Armed Bandit is actually a sequential decision model, wherein the objective entails the selection of actions in a stepwise manner to optimize the overall reward acquired over time [1]. In the field of recommender systems, the arm is usually recommendable content. And the reward obtained is

DOI: 10.54254/2755-2721/34/20230322

usually whether the user clicks or not, how satisfied the user is with the recommended content etc. However, a dilemma of exploration and exploitation will appear. Assuming that the total number of recommendations is a constant n, whether the arm choose each time is the best one explored so far needs the determination, for temporary short-term rewards, or whether it will be used to "explore the potential" of other arms and improve the knowledge of all of them, in order to maximize long-term rewards.

Currently, there are three main types of algorithms that researchers can weigh this exploration and utilization. The first is the Explore-Then-Commit (ETC) algorithm [2]. Its main idea is: first explore each arm the same number of times, and after that, based on the performance of each arm in the exploration phase, choose the best-performing arm as the only choice to be recommended afterwards. The second is the Upper Confidence Bound (UCB)algorithm, which maintains two values for each arm, A and B. A represents the mean value of the rewards that the arm has received so far, and B is an equation representing the number of times the arm has been explored [3]. When the arm has been explored fewer times, the value of B is larger for that arm. And the arm can be chosen with the largest A+B value money every time. The third one is Thompson Sampling (TS) algorithm [4]. The fundamental concept involves maintaining a probability distribution for each arm, where the rewards of each arm are assumed to follow this distribution. When selecting an arm, a random value is drawn from the current distribution of each arm, and the arm yielding the highest value is chosen. Subsequently, the distribution of the selected arm is updated based on the rewards obtained in that round.

The Multi-Armed Bandit algorithm used in recommender systems is basically the UCB algorithm. And in real performance, TS algorithm usually performs better on average. This aspect represents a potential area for enhancement, while recommender systems continue to encounter various challenges. These challenges include ensuring recommendation diversity, addressing the cold start problem, optimizing algorithms, and more [5].

The Multi-Armed Bandit algorithm has attracted a great deal of attention in industry, especially in the last few years with the majority of publications. And the role of these algorithms in the recommendation domain is also very important. While there are many articles describing the Multi-Armed Bandit algorithm, there are few literature reviews on the applications of these algorithms to the fields which Multi-Armed Bandit algorithm can use on. For the utilization of Multi-Armed Bandit algorithm in various of systems, the article will provide a review in the following order: 1) introduction to three classical Multi-Armed Bandit algorithms 2) Introduction to three application fields of Multi-Armed Bandit algorithm 3) a summary of this review paper.

2. Multi-Armed bandit algorithms

2.1. Explore-then-commit (ETC)

This research commences with an exploration of the most elementary algorithm applicable to the problem at hand, wherein the distribution of rewards for each arm remains unknown. A logical approach involves initiating an algorithm that explores each arm for a specific number of instances, hereby referred to as the exploration phase. After this, the arm with the largest mean reward based on the results of the exploration phase was chose, and in all subsequent times, this arm was selected to receive the reward. Supposing there are k arms, and in the exploration phase,e each arm is explored m times, and A_t denotes the arm chose for the t time. Then the formula for A_t is as follows.

$$A_t = \begin{cases} (t \bmod k) + 1, & if\ t \leq mk \\ argmax_i\ \hat{\mu}_i(mk), & t > mk \end{cases} \quad (1)$$

In terms of the ETC algorithm, it is definitely able to perform better than the random algorithm. Because it first tried to find the best arm. But its shortcomings are also obvious. For example, when there are many arms, the exploration phase will occupy most of the total number of times, which makes the number of times to exploit the best arm too few. More details of the ETC algorithms can be found in [6].

DOI: 10.54254/2755-2721/34/20230322

2.2. *Upper confidence bound (UCB)*

The steps of the UCB algorithm are as follows. For each arm, the algorithm maintains a UCB value for it. Here, the UCB value is the estimated value mentioned earlier. Suppose that the UCB value of the arm i is represented by UCB(i), And so far, after t rounds, the UCB value of arm i at round t+1 is as follows:

$$UCB(i) = \hat{\mu}_{i,N_i(t)} + \left(\frac{\alpha \log t}{2\, N_i(t)}\right)^{\frac{1}{2}} \tag{2}$$

$N_i(t)$ represents the number of times arm i is selected within t rounds. The first term represents the mean of the rewards that arm i has received so far. If arm i is more excellent, then the mean obtained by shaking it is more likely to be large, and the first term of arm i is larger. Therefore, the first term represents "Exploitation". It can be observed that when the number of times arm i is selected increases, the denominator in the radical will increase, and then the second term of UCB(i) will decrease. Conversely, if doing not choose arm i, then the numerator in the second term in the root of UCB(i) is increasing and the denominator is unchanged. This represents an increase in its uncertainty, so the second term is larger. In fact, the second term of the UCB value represents the upper bound of the possible true value of arm. So, the second term represents "Exploration". α is the exploration coefficient. If α is increased, the algorithm is more inclined to select the arm with fewer exploration times, that is, it is inclined to "Exploration". If α decrease, algorithm tends to explore the arm with higher mean to obtain stable revenue, it is likely to favor "Exploitation". Initially, the algorithm sets the mean of all arms to be the same. If prior knowledge is available, adjustments can be made.

2.3. *Thompson sampling (TS)*

The origins of this algorithm can be traced back to a paper from the 1930s [7]. This paper describes a sampling strategy in Bayesian methods for making decisions under an unknown probability distribution. And TS algorithm is also based on this theory. The TS algorithm "assumes" that each arm is subject to some distribution, and this distribution is constantly updated based on the sampled values.

It can be assumed that each arm follows a beta distribution. First, the algorithm initializes both parameters a, b of the beta distribution of each arm to 1. Subsequently, during each round of the algorithm, random sampling is performed from the beta distribution associated with each arm, and the arm corresponding to the distribution with the highest sampled value is selected. Subsequent to arm selection, the parameters of the beta distribution associated with the chosen arm are updated based on the obtained rewards. The parameter updating process is contingent upon the specific context of the application. For example, in the case of a binary classification problem, wherein the outcomes are limited to success or failure, the parameter "a" of the beta distribution is incremented by 1 upon achieving success, while the parameter "b" is incremented by 1 upon experiencing failure.

In general, the beta distributions for different arms end up being quite different depending on their distributions, as shown in Figure 1 below:

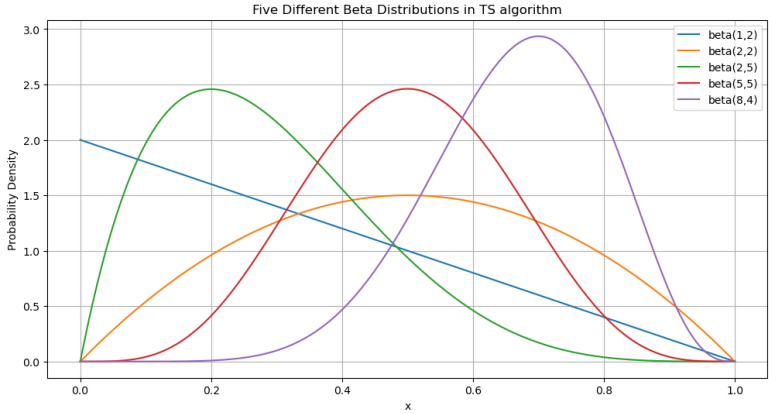

Figure 1. Five Different Beta Distributions (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/34/20230322

3. Applications of multi-armed bandit algorithms

3.1. Cold-start user recommendation

The cold-start problem in recommender systems pertains to the challenge of making accurate recommendations to users when their preferences are entirely unknown [8]. In this case, the validity of empirical evaluation would be greatly reduced. It performs well to transform the problem of mode selection for recommender systems into Multi-Armed Bandit system, which can operate without prior knowledge. Accordingly, an approach to tackle the cold-start problem involves employing the MAB model during the initial phase and subsequently transitioning to more effective models as the knowledge of users progressively accumulates.

While this approach has demonstrated favorable outcomes, its exploration remains somewhat limited. In this paper, only UCB1 algorithm and its variants are used and compared. The next research direction could be to improve using variants of the TS algorithm. At the same time, there is a lack of research on user coverage and does not probe the algorithm performance when recommending to a large number of users.

3.2. Portfolio selection problem

Portfolio selection is a classical problem in the field of finance and investment, which involves how to reasonably select and allocate different asset portfolios in order to minimize the risk and achieve the expected return goal. In investment, investors are often faced with a variety of different investment options. Each asset has its specific risk and return characteristics. The goal of investors is to reasonably select an asset portfolio according to their risk tolerance and income expectation, so that the overall portfolio can obtain the optimal expected return under the given risk level.

Usually, Markowitz's mean-variance optimization is seen as the best model to solve such problem. However, due to the difficulty of parameter estimation and other reasons, in some cases, it is more appropriate to reduce this problem to a Multi-Armed Bandit problem. In short, this approach is to treat each asset as an arm and adopt the TS algorithm strategy, and incorporate a return function based on the user's different investment risk preferences [9].

3.3. Selecting state machine policies for robotic system

Generating an optimal policy for a singular instance proves to be a challenging task for robots, primarily due to the prohibitive costs associated with evaluating such policies. As a result, the prevailing approach in this context is to focus on learning policies that exhibit satisfactory performance on average [10].

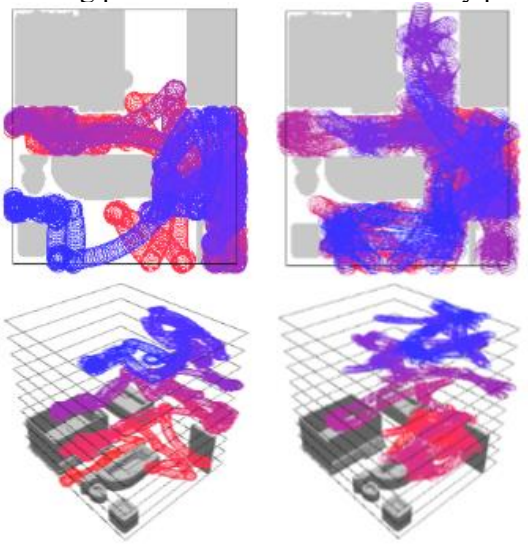

Figure 2. Motion time plots of the state machine for two different strategy choices in a narrow space [10].

DOI: 10.54254/2755-2721/34/20230322

As shown in Figure 2. Motion time plots of the state machine for two different strategy choices in a narrow space, there are two state machines acting within the same room layout, with time represented by vertical axis from bottom to top, or color from red to blue [10]. The state machine on the left uses large sweep curves, which perform well in large Spaces, but poorly in small Spaces. In contrast, the state machine on the right, which is suitable for navigating tight Spaces, performs better in this room. The purpose of this example is to prove that it is difficult to efficiently generate an optimal policy for a single example.

To solve this problem, the problem can be reduced to a k-state multi-armed bandit, where each state machine can be viewed as an arm. At each run, an arm is randomly selected and pulled, and the reward is randomly obtained according to the distribution of the arm. However, there is a downside to this approach: if there are too many state machines, the process will also take too long due to trying all the arms at once at the beginning. The excessive number of arms is also the difficulty encountered by the Multi-Armed Bandit Problem in most cases. In order to solve this difficulty, the combination of multi-armed bandit and collaborative filtering can be considered [10, 11].

4. Conclusion

This paper analyzes the Multi-Armed Bandit algorithm which has wide applicability and potential at present and investigates some fields of Multi-Armed Bandit application. In the explanation part of the algorithm, this paper mainly explains the logic of ETC, UCB, TS three algorithms. The performance of these three algorithms, from the perspective of regret value analysis, is successively better. However, UCB algorithm is still the most widely used algorithm, and TS algorithm is not as widely used in commercial fields due to its lack of interpretability and instability. Furthermore, this paper also explains the Multi-Armed Bandit algorithm in three aspects: cold start of the recommendation system, investment, and robot route decision. At the same time, it also shows the shortcomings of Multi-Armed Bandit algorithm in these three fields and the prospect of future development. However, this paper still has some limitations. For example, the theoretical explanation of the three Multi-Armed Bandit algorithms is not deep enough, and there is no quantitative analysis performance. The latest algorithms are not included, and more application areas of Multi-Armed Bandit algorithms are not introduced. These problems should be solved in the future.

References

[1] Slivkins A 2019 Introduction to multi-armed bandits. Foundations and Trends in Machine Learning 12.1-2 1-286
[2] Garivier A et al 2016 On explore-then-commit strategies Advances in Neural Information Processing Systems 29
[3] Garivier A and Eric M 2011 On upper-confidence bound policies for switching bandit problems International Conference on Algorithmic Learning Theory. Berlin, Heidelberg: Springer Berlin Heidelberg
[4] Agrawal S and Navin G 2012 Analysis of thompson sampling for the multi-armed bandit problem Conference on learning theory. JMLR Workshop and Conference Proceedings
[5] Ravi A N 2020 Unreliable multi-armed bandits: A novel approach to recommendation systems 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS) IEEE
[6] Lattimore T and Csaba S 2020 Bandit algorithms Cambridge University Press
[7] 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.
[8] Felício C Z et al 2017 A multi-armed bandit model selection for cold-start user recommendation Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization
[9] Zhu M et al 2019 Adaptive portfolio by solving multi-armed bandit via thompson sampling arXiv preprint arXiv:1911.05309

DOI: 10.54254/2755-2721/34/20230322

[10] Matikainen P et al 2013 Multi-armed recommendation bandits for selecting state machine policies for robotic systems 2013 IEEE International Conference on Robotics and Automation IEEE
[11] Su X Y et al 2009 A survey of collaborative filtering techniques Advances in artificial intelligence 2009