Improvements of Multi-Armed Bandit Algorithms

Hengnian Pan

{hengnian@uw.edu}

College of Arts and Sciences, University of Washington, Washington, 98195, United States

Abstract. As a branch of reinforcement learning, Multi-Armed Bandit(MAB) is used in problems of making optimal decisions such as recommendation and clinical trials. It has the ability of gathering information from unknown environments and using this information to maximize the reward in a process. This paper overviews the paths to further improve this algorithm and identifies two major methods of improvement. The internal method focuses on changing the inner structure of the algorithm to make it more generalizable in various circumstances; the external method instead focuses on constructing multi-phase processes with MABs and other algorithms to utilize MAB in more complex tasks. Both paths let the resulting algorithms show better performances than those before improvements in many different realms. Depending on the application, there are many other factors in specific occasions that may influence the performance of MAB, and targeted adjustments may also be needed.

Keywords: Reinforcement learning; Multi-armed Bandit; recommendation; clinical trials

1. Introduction

Reinforcement learning is a branch of machine learning where the algorithm is trained by the cycle of actively making trials, receiving feedback from them, and improving the decision-making process accordingly. It is applied when machines need to make decisions with little human intervention, such as in chess games, automatic driving, or personalized recommendations. Multi-Armed Bandit(MAB) is a special case of reinforcement learning, where the state does not change based on the learner's action. In this algorithm, each action is set as an arm, and pulling each of them yields different rewards. The goal is to maximize the total reward in a run. At the beginning, there is little information available, so it is important to learn the distribution of reward in each arm by pulling the unknown ones(exploration), while obtaining more reward by pulling the known arms with high reward(exploitation). MAB algorithms are designed in such a way that they are able to simultaneously balance these two types of action in order to maximize the reward at the end.

The application of MAB is wide. Because of its ability to explore from the outcome, it is commonly used in recommendation systems, where the program can analyze customers' behavior regarding the products shown, and adjust the personalized recommendation to show products more likely to attract people to purchase. This process involves the exploration-

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365102

exploitation sequence: utilizing information gained from previous customer behaviors to maximize reward while attempting to obtain a more comprehensive view of users' preferences[1]. Similarly, MAB can be used in dynamic pricing problems, where the decision-making with uncertain rewards fits in the mechanics of pulling arms[2], or clinical trial problems, where different treatments and their effects can also be quantified as arms and rewards[3].

MAB naturally fits into many real-world problems, while the basic MAB algorithms usually need to be further improved, as the cases in reality are generally more complicated than what they assume. There are various ways of improvement: it can be modified from inside, which provides additional traits for an MAB to fit in specific environments, such as contextual or non-stationary rewards; it can also be combined with other algorithms, forming a process with multiple phases, to be applied in more complicated cases. These specified modifications enable MAB to employ more of its potential in various applications. This paper introduces the two types of modification with some widely applied examples and real cases where the changes make improvements.

2. MAB Algorithm

The basis of the Multi-Armed Bandit algorithm is the set of arms, and pulling an arm yields a random reward. Each arm has a distribution of rewards which is unknown at the beginning of an experiment. The ultimate goal is to maximize the total reward after the experiment. However, the concept of maximizing reward is difficult to analyze. The growth of total reward varies in different experiments, so it is hard to measure what is a good amount of reward after pulling an arm. Instead, researchers usually apply the analysis on regret, which is the difference between the reward gained and the best possible reward. From this perspective, when the regret is zero at a pull, the learner knows it pulls the optimal arm. With the definition of regret, the main goal of MAB can be interpreted as minimizing the cumulative regret.

In the process of an MAB run, the core problem is the tradeoff between exploration and exploitation. For the aim of minimizing regret, the algorithm should pull the optimal arm as many times as possible. However, as the reward of every arm is unknown at the beginning, the algorithm needs to pull these unknown arms as well in order to obtain information about their reward distribution. This yields the exploration-exploitation dilemma: on one hand, if the exploration takes too long, too many pulls are wasted on gathering information from suboptimal arms; on the other hand, if the exploration phase is too short, the algorithm is less certain about the reward distribution of each arm, making it harder to select the optimal arm. An effective MAB algorithm should balance the time for both processes, obtaining the amount of information just right to determine the optimal arm and pulling this arm as much as possible.

There are two main types of MAB: deterministic and stochastic. Upper Confidence Bound(UCB) uses the information gathered from arms to establish confident estimations on the mean reward of each arm; Thompson Sampling(TS) updates the estimation of the reward distribution of the arm after each pull. Both UCB and TS can utilize the information to do their best in exploring the optimal arm, and are able to reach logarithmic growth of regret in a standard environment. Nevertheless, to perform ideally, these basic MAB algorithms need a considerable number of assumptions, such as independent arms, sub-Gaussian reward distributions, and stationary distributions. In the regret calculation, these conditions are required to bound the growth of regret to be sub-linear. In the real world, though, many application cases cannot assure these conditions, thus the basic MABs may fail. Another limit is about the scale. Sometimes the

overall task could be too complicated or involve too many phases to fit into one arms-rewards relationship, so that using MAB individually is unable to solve the problem.

3. Directions of Improvement

There are two major ways to improve the utility of MAB in order to allow it to cover more types of tasks. The first is changing the internal mechanics, such as the method of choosing arms and calculating rewards. This method helps MAB adapt to cases where some of the assumptions are not true. The second is dividing the problem from outside, where an individual MAB can handle a part of the whole problem and combine the work with other parts done by other algorithms, which may or may not be MABs.

3.1 Modification from inside

Classic MAB algorithms need to make several assumptions in order to reach a bounded regret, while there are still solutions when these assumptions are not met. These modified MABs are designed to adapt to certain non-ideal circumstances and are able to maintain good performance when the basic ones lose their effect.

Adversarial bandit. The regret calculation for basic stochastic MAB is largely dependent on the assumption that the reward distribution of all arms is sub-Gaussian. In many cases, this is valid, as Gaussian distribution and bounded distribution, which is guaranteed to be sub-Gaussian, are indeed very common in the real world. On some occasions, though, when researchers want a more generalizable result in a more unpredictable environment, it may be better to not make such an assumption. Adversarial MAB comes into place when there is no assumption made for the reward. This type of bandit, such as Exp3 and its variants,is designed to focus more on exploration and react faster when observing a change in collected reward distribution. A simulation was made about pedagogical experiments, where teaching conditions are arms and students' performances are rewards. It was found that the reward of the Exp3 family remains stable despite the change in standard deviation of arms, while the reward of TS significantly drops as the standard deviation of arms increases[4]. When the reward distribution follows the assumption of being sub-Gaussian, the performance of adversarial bandits can be worse, but in an unknown environment, adversarial bandits are a more robust choice.

Contextual bandit. Another important assumption made in basic MAB algorithms is that the reward of arms is independent of any external factors. However, in real world experiments, this is usually not the case. For example, in clinical trials testing the efficacy of various treatments, it is inappropriate to assume that all patients respond identically to a treatment. It is proven that the performance of a treatment can vary based on different body conditions of patients[3]. This is when contextual bandits become useful. It is a type of modified MAB where an external context variable is added to the reward distribution of each arm. Before each pull, the reward distribution of arms changes based on the context, in this case the body conditions of each patient. Then, the optimal arms are calculated, which may be different under different contexts. This way, the algorithm is able to dynamically classify patients and apply pertinent treatments to each one. In an experiment on treatments for ischemic strokes, the contextual TS method reaches a 72.63% probability of applying the best treatment on each patient, being significantly better than 64.34% in context-free TS bandit[4].

Non-stationary bandit. Classic MAB algorithms perform well in a stationary environment, where the reward of arms does not change over time, but the world is ever-changing. For instance, the demand for goods changes because of market activities; the effectiveness of medicine changes as a result of pathogen mutation. Consequently, when applying MAB algorithms in those environments, the classic implementations of bandit may no longer be able to limit the regret in a reasonable growth rate[5]. This problem can be addressed by enabling the algorithm to "forget" the information that is too old. A common solution is adding a "discounded(DS)" or "sliding window(SW)" feature to the TS or UCB bandit. The DS strategy considers earlier results less important and discounts their weight, and the SW strategy completely drops the results if they are not recent enough[5, 6, 7]. These algorithms only take recent results into account, the arms that are not observed for many rounds will have their estimated distribution reset and available to be explored again.

3.2 Combination from outside

When one MAB is unable to complete a task, it is beneficial to add other algorithms from outside. The choice of addition is flexible: it can be other MABs or different algorithms that better suit the rest of the task.

Multiple MAB phases. In complicated applications where a single MAB is unable to cover all the procedures, a multi-phase MAB can be applied. One of such scenarios is the application of a reconfigurable intelligent surface(RIS) and mmWave transmitter(TX) in information transmission. Along with the establishment of communication, the phase shift of RIS and TXneeds to be simultaneously fine-tuned. Hence, it is appropriate to apply a MAB algorithm on each of them and alternately adjust their phase shift[8]. This method demonstrates a close to optimal performance even in harsh blockage environments.

Another example is online recommendation, where using individual items as arms results in an extremely large arm set, while using categories as arms cannot yield an accurate outcome. A large algorithm with two nested MAB phases is thus developed: in the first step, the items are dynamically distributed into different groups as arms, and after selecting the best few groups, the second step uses items in each group as arms, finally picking the result for the recommendation list. Compared with single phase MAB, this approach raises the performance by optimally nearly 50%[1].

MAB with other algorithms. MAB can also accomplish a partial task in a larger project. By combining with other types of algorithms into a hybrid method, it is able to complement the shortcoming and make the overall process more effective. In federated learning, where the server lets local users train models and integrate them together to the global model, it is important to distinguish free-riders, who do not train their models well, thus lowering the quality of the outcome. MAB fits well for this task by setting local trainers as arms and the quality of trained sample models as reward[9]. This way, the contributive trainers can be properly selected before the second step which uses an auction process to distribute training opportunities.

In a recommendation with various objectives, i.e. the goal is not only increasing customer response, but also ensuring the profit of companies, a Juggler framework is used to choose the optimal degree of compensation and relevance(low, neutral, high). However this framework depends on five pre-determined settings and has little ability to make quick responses to changes in the environment[10]. In this scenario, MAB is also a good additional component to fine-tune the parameters in a timely manner. As a result, this hybrid method, compared with using Juggler

framework solely, reduces regret by 13.7% and increases the chance to make optimal choices by 9.8%[10].

4. Other Potential Challenges

Besides these improvements that are able to deal with specific conditions, there are more potential challenges that may exist when applying MAB in practice. These are some examples that are not classified into the previous challenges.

4.1 Interference

In real life, when researchers convert a problem into bandits, there sometimes need to be multiple experiments proceeding in parallel in order to better utilize the space. For example, when medical workers want to test the effect of different policies in controlling COVID-19, they may use each community as an individual unit of the MAB algorithm, and each unit tests arms separately. However, the communities are not totally isolated, because people transit between them and so do pathogens. This means that a policy used in one of them also affects nearby communities, making the performance of this policy dependent on the performance in other ongoing experiments[11].

4.2 Cold Start

The ability of MAB in making decisions based on previous information is strong. Nevertheless, there is usually a special stage in these algorithms: when a run has just started, there is little information available. A common case in online recommendation is when a new user signs up on a platform, and no record of this user's actions can be taken into account. In this circumstance, one choice is to offer totally random options, which requires more time to begin personalized recommendations; another choice is to offer items that are more popular among other users, but this approach loses the accuracy of determining the new user's preference[12]. It is challenging to balance the approach and address both problems.

5. Conclusion

Multi-Armed Bandit is a strong algorithm, while it has a great potential of enhancements. This paper is an overview of the two major types of improvements on MAB to make them more effective. The first type is modifying the internal structure of the exploration-exploitation sequence in order to fit the MAB in more complex environments. This is mainly done by removing some assumptions on the environment in classic MAB and accordingly adjusting the reward calculation and arm selection process to maintain an optimal regret growth. The second type is combining with other algorithms to form multi-phase processes. The choice of combination is wide, which could be other MAB stages or other suitable algorithms. This way the advantage of MAB may complement that of others to solve a problem which requires more steps. This is not a comprehensive list, as there are even more possible shortcomings of MAB which could affect its performance under certain conditions. As an important branch in reinforcement learning, it has a wide range of applications in various realms, so the interfering factors could appear in different forms from the actual problem it is applied on. It is important to correspondingly upgrade the algorithm to maximally utilize its potential.

References

[1] Yan, C.; Han, H.; Wang, Z.; Zhang, Y.: Two-phase multi-armed bandit for online recommendation. Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 1–8 (2021)

[2] Tajik, M.; Mohamadpour Tosarkani, B.; Makui, A.; Ghousi, R.: A novel two-stage dynamic pricing model for logistics planning using an exploration–exploitation framework: A multi-armed bandit problem. Expert Systems with Applications. p. 123060 (2024)

[3] Varatharajah, Y.; Berry, B.: A contextual-bandit-based approach for informed decision-making in clinical trials. Life (Basel). p. 1277 (2022)

[4] Yang, Z.H.; Zhang, S.; Rafferty, A.N.: Adversarial bandits for drawing generalizable conclusions in non-adversarial experiments: An empirical study. Proceedings of the Educational Data Mining Conference (2022)

[5] Trovò, F.; Paladino, S.; Restelli, M.; Gatti, N.: Sliding-window Thompson sampling for non-stationary settings. ArXiv preprint (2020)

[6] Qi, H.; Guo, F.; Zhu, L.: Thompson sampling for non-stationary bandit problems. Entropy. p. 51 (2025)

[7] Askhedkar, A.R.; Chaudhari, B.S.: Multi-armed bandit algorithm policy for LoRa network performance enhancement. Journal of Sensor and Actuator Networks. p. 38 (2023)

[8] Mohamed, E.M.; Hashima, S.; Hatano, K.; Aldossari, S.A.: Two-stage multi-armed bandit for reconfigurable intelligent surface aided millimeter wave communications. Sensors. p. 2179 (2022)

[9] Lu, R.; et al.: Two-stage client selection for federated learning against free-riding attack: A multi-armed bandits and auction-based approach. IEEE Internet of Things Journal. pp. 33773–33787 (2024)

[10] Cunha, T.; Marchini, A.: A hybrid meta-learning and multi-armed bandit approach for context-specific multi-objective recommendation optimization. ArXiv preprint (2024)

[11] Xu, Y.; Lu, W.; Song, R.: Linear contextual bandits with interference. ArXiv preprint (2024)

[12] Silva, N.; Silva, T.; Werneck, H.; Rocha, L.; Pereira, A.: User cold-start problem in multi-armed bandits: When the first recommendations guide the user's experience. ACM Transactions on Recommender Systems (2022)