

Multi-Armed Bandits: Algorithms, Applications, and Future Directions

Yurong Zheng *

The Pennsylvania State University, Pennsylvania, The United States

* Corresponding Author Email: 2682677617zyr@gmail.com

Abstract. The Multi-Armed Bandit (MAB) problem is a fundamental framework in sequential decision-making. The main goal is to balance exploration of uncertain options with exploitation of known rewards. Originating from classical probability theory, MAB has evolved into a significant modern reinforcement learning. This problem can be applied to all walks of life in reality, from online advertising and recommendation systems to healthcare and finance. Over the years, algorithms such as Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), and Thompson Sampling (TS) have emerged as leading strategies, each offering unique trade-offs between exploration efficiency and theoretical regret bounds. This paper gives an overview of the history, main algorithms, applications, and challenges of MAB, and discusses possible future research directions. Furthermore, it highlights the practical implications of these algorithms for real-world decision-making scenarios.

Keywords: Multi-Armed Bandit, Thompson Sampling, Reinforcement Learning.

1. Introduction

In many real-life situations, we are often required to make decisions without having complete knowledge of all possible outcomes. The Multi-Armed Bandit (MAB) problem provides a formal framework to model this type of uncertainty. In this setting, an agent must repeatedly choose among several options, or "arms," each of which provides a reward drawn from an unknown probability distribution. The agent faces a fundamental trade-off: it must decide when to explore less-known arms to gather information and improve future decisions, and when to exploit the arm that currently appears to yield the highest reward. The main objective is to minimize regret, which measures the difference between the cumulative reward actually obtained and the reward that could have been obtained by always selecting the optimal arm. This problem has widespread applications in areas such as online recommendation systems, digital marketing, personalized medical trials, and A/B testing for product features. Since data is often limited, decisions can be costly, and user preferences may change over time, it is crucial to design efficient algorithms that balance exploration and exploitation while adapting to dynamic environments. Various strategies, including Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), and Thompson Sampling (TS), have been proposed to address these challenges, providing theoretical guarantees and practical performance improvements.

2. Development of Bandit Algorithms

2.1. Early Work

Robbins (1952) first studied the bandit problem. Early methods often used fixed exploration schedules, which were simple but not very efficient [1].

2.2. Explore-Then-Commit (ETC)

ETC splits the process into two stages: first explore each arm a set number of times, then pick the best one for the rest of the rounds. In the first stage, called exploration, each arm is pulled the same number of times, say m . The algorithm records the average reward of each arm:

$$\mu_i = 1/m \sum_{\{t=1\}}^m X_{\{i,t\}} \quad (1)$$

After exploration, the algorithm enters the commit stage. It chooses the arm with the highest estimated mean, and keeps pulling this arm for the rest of the rounds. The method is simple and gives basic guarantees, but it cannot adjust if the situation changes [2].

2.3. Upper Confidence Bound (UCB)

UCB algorithms choose the arm with the best optimistic estimate. They use both the average reward and a confidence term that decreases with more samples. For each arm i , the algorithm calculates an estimate of its mean reward plus a confidence term:

$$UCB_i(t-1) = \mu_i(t-1) + B/2 \sqrt{\{4 \log n\}/\{T_i(t-1)\}} \quad (2)$$

UCB achieves logarithmic regret and is widely studied because of its strong theory and good performance.

2.4. Thompson Sampling (TS)

TS is a Bayesian method. It samples from the estimated reward distribution of each arm and picks the arm that looks best in that sample. In the common Gaussian version, the algorithm samples:

$$F_i(t) \sim N(\mu_i(t), \{B^2\}/\{4T_i(t)\}) \quad (3)$$

This allows natural balance between exploration and exploitation. TS often performs very well in practice [3].

3. Applications

3.1. Advertising and Recommendation

One of the most common uses of the Multi-Armed Bandit problem is in online advertising and recommendation systems. Companies often need to choose which advertisement, article, or movie to show to a user. If the choice is random, the click-through rate or engagement will be low. If the system always shows the same option, it may miss better ones. Bandit algorithms help balance these two extremes.

For example, a video platform can treat each movie genre as an arm. When a user logs in, the system chooses one genre to recommend. Over time, it learns which genres bring higher ratings or longer watch times. T. Geng et al. (2021) from Stanford University state that “We compute the value of adaptive design ... the adaptive design has been able to significantly cut down the opportunity cost of experimentation for the advertisers.”

3.2. Healthcare

Healthcare is another area where bandit algorithms have strong impact. In clinical trials, doctors test different treatments on patients to see which works best. Traditional trials assign patients randomly and wait until the end of the study to choose the best drug. This may expose many patients to weak or harmful treatments [4].

With a bandit approach, the system learns as the trial goes on. If one drug shows better results early, more patients are assigned to it. As M. Aziz et al. (2021) wrote in their research, “Motivated by the literature on multi-armed bandit models, we advocated the use of the powerful Thompson Sampling principle ... significantly reducing the allocation to doses with high toxicity.” [2] This reduces the number of patients who get poor treatments and speeds up discovery of the most effective option.

3.3. Finance:

Financial decisions often involve balancing risk and return. This makes finance a natural field for bandit algorithms. In portfolio optimization, each investment option can be seen as an arm. The investor wants to explore different assets but also exploit the ones that give good returns [5].

For example, a trading system may use UCB to test new stocks while still focusing on known profitable ones. Bandit approaches allow investors to adjust quickly to market changes, rather than relying only on historical data.

4. Challenges and Extensions

4.1. Contextual Bandits:

Many problems depend on user or environment features. In many real applications, decisions are not only about which arm to pull, but also about the context, such as user features or environment information. For example, a movie recommendation system needs to know not only which movies are available, but also who the user is, or what device they are using. Adding this context makes the model more realistic and powerful, but also much harder to solve [6].

Luo et al. (2018) studied contextual bandits in non-stationary worlds. They proposed methods to detect when the environment changes and restart the algorithm to adapt. This shows that contextual bandits can be applied in real settings where user interests or external conditions are constantly shifting [7].

4.2. Non-Stationarity

In many domains, the reward distribution may change over time. User preferences may change over time, new products may appear, and seasonal effects may change behavior. If the algorithm assumes that the best arm never changes, it may perform poorly after the environment shifts.

Di Benedetto et al. (2020) studied bandits in seasonal environments, such as music or movie recommendations where popularity changes with holidays or events.⁴ Their algorithm was able to detect seasonal changes and quickly adapt. This is important in practice because many applications cannot assume stability over long horizons.

4.3. Scalability

Large action spaces, such as in modern recommender systems, are difficult to handle with classic methods. In modern recommender systems or online platforms, there may be millions of possible actions and very high-dimensional context. Classic algorithms like ETC, UCB, or TS can become too slow or need too much memory. Scalable methods are needed to make bandits practical in industry.

Bouneffouf et al. (2017) developed contextual bandit algorithms that only use part of the context features, called “restricted context.”⁵ This makes the algorithm faster and more practical when the feature space is very large. More recently, deep learning methods have also been combined with bandits to handle large action spaces [8].

5. Future Directions

5.1. Multi-Objective Bandits

In many real problems, we need to balancing more than one goal such as accuracy and diversity. For example, in recommendation systems, we may want to maximize accuracy, but also keep diversity so users do not see the same kind of items all the time. Multi-objective bandits try to balance more than one goal at once. This means the algorithm must decide how to trade-off between objectives. It is a difficult problem, because improving one goal may hurt another.

5.2. Robust Bandits

Making methods work well even when models are imperfect or under attack could be a problem. Standard bandit algorithms usually assume that the model and data are correct. While Robust bandits aim to keep good performance even when the environment is not perfect. This includes handling unexpected changes, errors, or attacks on the system. The idea is to make the algorithms more stable and reliable, so they can be trusted in more difficult situations.

6. Conclusion

The Multi-Armed Bandit problem is a key model for learning under uncertainty. From simple methods like ETC to advanced approaches like UCB and TS, MAB has influenced many real applications. While there are challenges with large-scale, dynamic, and fair decision-making, research continues to expand, making MAB a rich and active field.

References

- [1] Geng, T., Li, X., Nair, H. S., Hao, J., Xiang, B., & Fan, S. (2021). Comparison Lift: Bandit-based experimentation system for online advertising. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35 (13), 15117–15126.
- [2] Aziz, M., Kaufmann, E., & Riviere, M. K. (2021). On multi-armed bandit designs for dose-finding clinical trials. *Journal of Machine Learning Research*, 22 (14), 1–38.
- [3] Luo, H., Wei, C.-Y., Agarwal, A., & Langford, J. (2018). Efficient contextual bandits in non-stationary worlds. *Proceedings of the Conference on Learning Theory*, 75, 1739–1776.
- [4] Di Benedetto, G., Bellini, V., & Zappella, G. (2020). A linear bandit for seasonal environments. *arXiv preprint arXiv:2004.13576*.
- [5] Bouneffouf, D., Rish, I., Cecchi, G. A., & Féraud, R. (2017). Context attentive bandits: Contextual bandit with restricted context. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 1468–1475.
- [6] S. Agrawal and N. Goyal. Further optimal regret bounds for Thompson Sampling. In *Proceedings of PMLR*, pages 39–46, 2013.
- [7] A. Garivier and T. Lattimore. On Explore-Then-Commit strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [8] E. Kaufmann, O. Cappe, and A. Garivier. On Bayesian upper confidence bounds for bandit problems. In *AISTATS*, pages 592–600, 2012.