A Comparative Analysis of Cumulative Regret Based on Multi-Armed Bandit Algorithms

Muqing Xue *

School of Economics and Management, Beijing Jiaotong University, Beijing, China

* Corresponding Author Email: 23711072@bjtu.edu.cn

Abstract. This study aims to conduct a detailed comparison of the performance of three classic Multi-Armed Bandit algorithms: Thompson Sampling, UCB, and ETC. The MAB problem, as an important sequential decision-making framework, primarily challenges lie in how to strike a balance between "exploration" and "exploitation". We quantitatively analyzed each algorithm's long-term performance using 100 independent experiments and cumulative regret as the primary metric. The experimental findings demonstrate that the three algorithms' performance varies significantly in the tested context. The Thompson sampling method performed the best, with the least increase in regret and the lowest final value. The UCB algorithm performed second-best, with regret growing logarithmically. The ETC algorithm saw rapid accumulation of regret in the early stages before stabilizing, but it had the poorest performance because it lacked the ability for continuous exploration. These findings confirm that the Thompson sampling method is the most efficient in balancing exploration and exploitation, and is the best choice for solving such Random Stationary Multi-Armed Bandit Problem.

Keywords: Multi-Armed Bandit, Cumulative Regret, Thompson Sampling, ETC, UCB.

1. Introduction

When dealing with uncertain decisions, such as in online advertising, we must make a trade-off between exploring the unknown and exploiting the known best options. The MAB problem provides an ideal model for this trade-off [1]. By reducing the cumulative regret, it seeks to assist us in determining the best course of action in a constrained number of cycles.

This study will conduct experiments to compare three major MAB algorithms: Thompson Sampling, UCB, and ETC. We will analyze how their respective exploration strategies affect the long-term performance, and quantitatively evaluate the differences in their performance on cumulative regret, providing data support for practical applications.

2. Random Stationary Multi-Armed Bandit Problem Definition

The random stationary Multi-Armed Bandit problem is a specific type of MAB, where the reward distribution of each arm is fixed and known, and does not change throughout the decision-making process. The core objective of this problem is to select the appropriate arms to maximize the long-term cumulative reward [2].

2.1. Mathematical Modeling

Suppose there are K arms in a MAB problem, where each arm has a reward distribution R_k, and the reward r_k for each arm is independently drawn from a fixed probability distribution with an expected return μ_k. In each decision round, the decision-maker selects an arm to pull and receives a reward r_k, which is drawn from the reward distribution of arm k.

To quantify the effectiveness of the decision strategy, we use two evaluation metrics: Cumulative Reward and Regret.

- **Cumulative Reward:**

The cumulative reward refers to the total reward accumulated by the selected arms over n trials. It is defined as:

$$C(n) = \sum_{t=1}^{n} r_{a_t}, \quad (1)$$

where r_{a_t} represents the arm selected at round t, and r_{a_t} is the reward of that arm.

- **Regret:**

Regret measures the difference in performance between the selected strategy and the optimal strategy. The optimal strategy chooses the arm with the highest expected reward, and regret is the difference between the reward of the optimal arm and the selected arm. It is defined as:

$$R(n) = \sum_{t=1}^{n} (\mu_{optimal} - \mu_{a_t}), \quad (2)$$

where $\mu_{optimal}$ is the expected reward of the optimal arm, and μ_{a_t} is the expected reward of the arm a_t selected at round t.

2.2. Characteristics of Random Stationary Setting

In the random stationary setting, the reward distribution for each arm remains constant throughout the experiment, meaning that the expected reward μ_k for each arm is fixed. Therefore, based on the known reward distribution, the optimal strategy for each arm can theoretically be computed. However, in practical applications, we can typically only estimate these expected rewards through exploration and experimental data. The goal of the algorithm is to select the optimal arm as much as possible within a limited number of exploration rounds, reducing regret and improving cumulative rewards.

2.3. Differences from Non-Stationary and Contextual Multi-Armed Bandit Problems

Unlike the random stationary MAB problem, in non-stationary environments, the reward distributions may change over time, making it more complex to optimize the decision strategy. This requires additional consideration of how reward distributions evolve over time. In the contextual multi-armed bandit problem, the reward distribution for each arm not only depends on its own characteristics but also on contextual information (such as user features), requiring the algorithm to dynamically select arms based on this contextual information. These problems increase the complexity of the decision process and often require more sophisticated algorithms.

In the random stationary MAB problem, since each arm's reward distribution is known and fixed, the decision process is relatively simpler. The main challenge is how to effectively explore within a limited time, quickly converge to the optimal arm, and maximize the total reward throughout the process.

3. Principles of Four Classic Algorithms

3.1. Thompson Sampling

Thompson Sampling is a Bayesian updating-based probability matching method. The core idea is to use Bayesian inference on the reward distribution of each arm and dynamically select the arm that is most likely to yield the maximum reward [3]. Specifically, Thompson Sampling assigns a prior distribution (usually a Beta distribution) to each arm, then updates the posterior distribution based on observed rewards, and selects the arm by sampling from the posterior.

Mathematical Derivation and Steps:

1. **Prior Distribution Setup:**

Assume the rewardr_k of arm k follows a known distribution, and we typically choose a Beta distribution as the prior:

$$\mathcal{P}(\theta_k) \sim Beta(\alpha_k, \beta_k), \quad (3)$$

where θ_k represents the success probability of arm k (i.e., the probability of receiving a reward of 1), and α_k and β_k are the shape parameters of the Beta distribution. Initially, all arms have the same prior distribution.

2. **Updating the Posterior Distribution with Observed Data:**

Each time arm k is selected and a reward $r_k \in \{0,1\}$ is observed, the posterior distribution for arm k is updated using Bayes' theorem:

- If r_k=1, update $\alpha_k \rightarrow \alpha_k$+1;
- If r_k=0, update $\beta_k \rightarrow \beta_k$+1.

The updated posterior distribution is:

$$\mathcal{P}(\theta_k|data) \sim Beta(\alpha_k', \beta_k'). \tag{4}$$

3. **Selecting the Arm Based on Posterior Probability:**

In each decision round, a value θ_k' is sampled from the posterior distribution of each arm, and the arm with the maximum θ_k' is selected:

$$a_t = arg\ max_k\ \theta_k'. \tag{5}$$

3.2. Explore-Then-Commit (ETC)

Explore-Then-Commit is a simple and intuitive strategy that separates the exploration phase from the exploitation phase. First, it explores by randomly selecting arms, and then after enough exploration, it selects the arm with the highest estimated reward for long-term exploitation. ETC is particularly useful for short-term tasks or scenarios with limited budgets [4].

Algorithm Steps and Mathematical Formula:

1. **Fixed Exploration and Exploitation Phases:**

Assume that during the first $T_{explore}$ rounds, arms are selected randomly to explore, and rewards are collected for each arm. After the exploration phase, the average reward $\hat{\mu}_k$ of each arm is computed, and the arm with the maximum average reward is selected for long-term exploitation:

$$a_t = arg\ max_k\ \hat{\mu}_k. \tag{6}$$

During the exploration phase, the arms are chosen randomly:

$$a_t \sim Uniform(1, 2, \dots, k). \tag{7}$$

During the exploitation phase, the arm with the highest estimated reward is selected:

$$a_t = arg\ max_k\ \hat{\mu}_k. \tag{8}$$

3.3. Upper Confidence Bound (UCB)

The Upper Confidence Bound algorithm balances exploration and exploitation by considering both the mean reward of each arm and the uncertainty of that mean. UCB aims to explore arms with higher uncertainty while exploiting arms with higher known rewards [4].

Algorithm Steps and Mathematical Formula:

1. **Calculate the Confidence Interval:**

Each arm's selection is based on its estimated reward $\hat{\mu}_k$ and the confidence interval for that arm's reward:

$$a_t = arg\ max_k \left(\hat{\mu}_k + \sqrt{\frac{2lnt}{n_k}} \right), \tag{9}$$

where $\hat{\mu}_k$ is the average reward of arm k, n_k is the number of times arm k has been pulled, and t is the current time step. This formula makes arms with greater uncertainty more likely to be explored.

2. **Theoretical Regret Bound:**

UCB has a theoretical regret bound that ensures, after T trials, the cumulative regret R (T) grows logarithmically:

$$R(T) = O(lnt). \tag{10}$$

This implies that UCB will converge to the optimal arm over time and maintain a small regret.

4. Performance Comparison and Experimental Analysis

In order to verify and compare the actual performance of the three classic MAB algorithms discussed in the previous review - Thompson sampling, UCB, and ETC - this study employed a real case study. We utilized the MovieLens 1M dataset and transformed it into a typical MAB problem, and evaluated the performance of each algorithm in terms of the cumulative regret metric.

4.1. Dataset and Problem Transformation

The MovieLens dataset contains 1 million pieces of data from approximately 6,000 users regarding their ratings of 4,000 movies, with the rating scale ranging from 1 to 5 stars [5]. To transform it into an MAB problem, the following mapping was carried out:

Round t: Each time the algorithm recommends movies to the user, it is regarded as one round.

Arm a: The movie type is regarded as the "arm" in MAB. To simplify the problem and improve computational efficiency, we randomly selected 10 movies as the 10 available arms for recommendation.

Reward: The user's rating is regarded as the reward. Since the MAB problem usually deals with binary rewards, we convert the user's rating into a binary reward: if the user rates the recommended movie with a score of 4 stars or higher, the reward is 1; otherwise, the reward is 0.

Through this transformation, our goal is to enable the MAB algorithm to, in consecutive recommendation rounds, by learning the popularity of different movies, maximize the total number of successful recommendations (i.e., cumulative rewards), thereby minimizing the cumulative regret.

4.2. Experimental Setup

We conducted independent experiments for each algorithm and standardized the key parameters:

- Total number of rounds: T = 100000.
- Number of repetitions: To reduce the influence of randomness on the results, each algorithm was run independently for 100 times, and the final results were obtained by taking the mean and standard deviation.

4.3. Results and Analysis

The following three graphs illustrate the changes in cumulative regret over the number of rounds for the three algorithms during 100 runs.

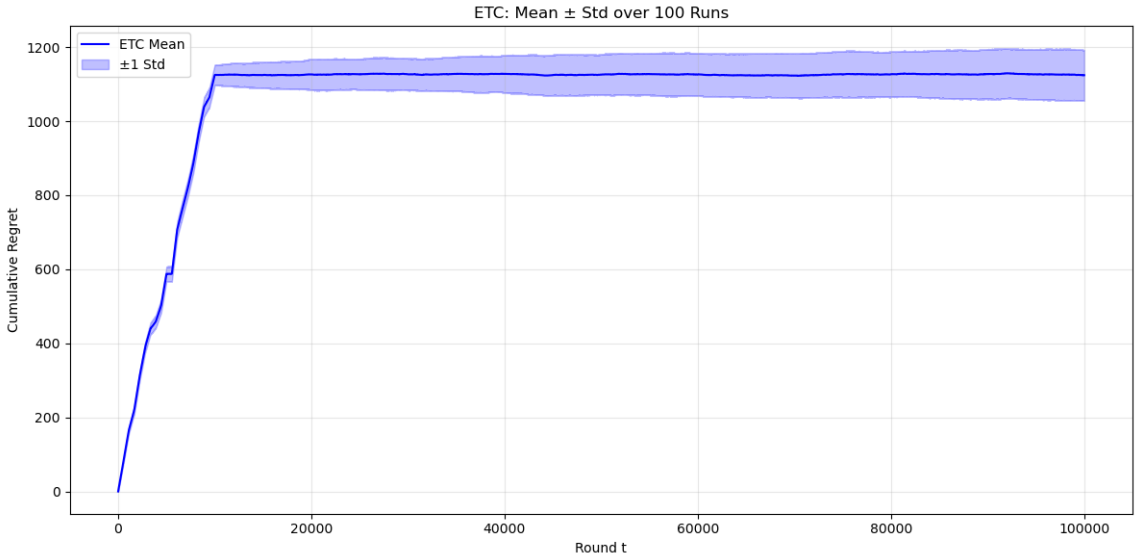


Fig. 1 Experimental results of the ETC algorithm

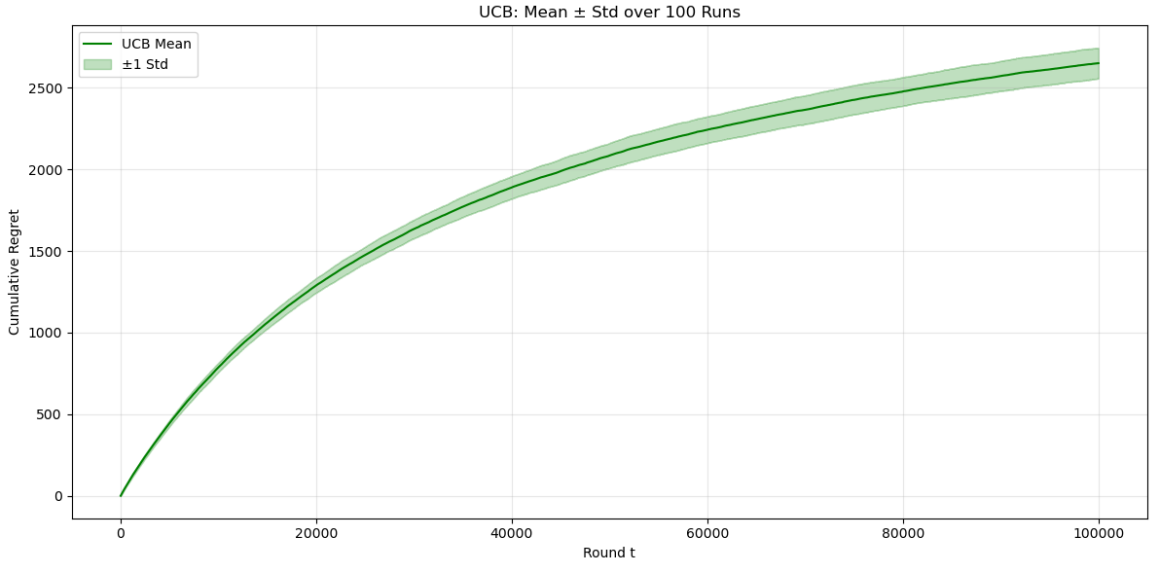


Fig. 2 Experimental results of the UCB algorithm

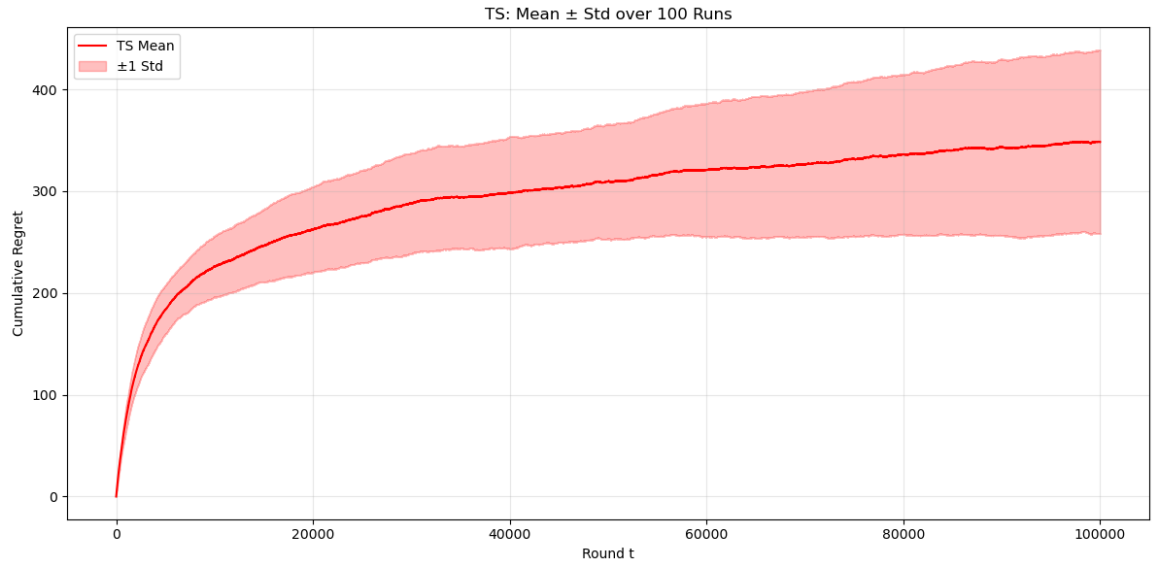


Fig. 3 Experimental results of the TS algorithm

From the experimental results, it can be seen that there are significant differences in the performance of the three algorithms. As shown in Fig. 1, experimental results of the ETC algorithm, the ETC algorithm experiences rapid accumulation of regret in the initial stage, and then the curve becomes relatively flat. Eventually, the regret is approximately 1150. This reflects its "exploration before exploitation" strategy: once the "best" movie genre is selected during the exploration stage, no further exploration will be conducted. However, if the initial judgment is incorrect, the regret cannot be corrected.

As shown in Fig. 2, experimental results of the UCB algorithm, the regret curve of the UCB algorithm keeps smoothly rising. The logarithmic growth of the regret indicates that UCB has been continuously and cautiously exploring. However, no optimal arm was found within the limited number of rounds.

As shown in Fig. 3, experimental results of the TS algorithm, it demonstrated the best performance. Its cumulative regret curve grew the slowest, with the final value being approximately 350. This once again confirms the effectiveness of the TS algorithm. It can achieve a more efficient and natural balance between exploration and exploitation through its probabilistic sampling mechanism, thereby minimizing the regret over the long term.

5. Discussion and Future Research Directions

Although this study has provided valuable insights into these algorithms, there are still many promising areas for exploration in the MAB field. Future research could focus on the following aspects.

5.1. Contextual Multi-Armed Bandit

The MAB model in this study assumes that the arms are independent and their reward distributions are fixed. However, in many real scenarios, rewards not only depend on the selected arm but also on contextual information, such as the user's age, gender, or historical behavior [6]. Future research can extend to contextual multi-armed bandit, integrating MAB with supervised learning techniques to leverage this contextual information for more accurate decision-making.

5.2. Algorithm Performance in Non-stationary Environments

Our experiments assumed that the environment was stationary, meaning that the reward distribution for each arm did not change over time. However, in many practical applications, the environment is non-stationary. For instance, the popularity of movies can vary over time. Future research could explore and compare the performance of these algorithms in environments where the reward distribution changes dynamically, and propose improved algorithms that can adapt to such changes.

5.3. The integration of MAB with deep learning

Traditional MAB algorithms usually rely on simple assumptions about the reward distribution or make decisions in a linear, simple context [7]. However, in many real-world complex scenarios, such as recommendation systems, advertising placement, and personalized medicine, the context information is often high-dimensional and non-linear. At this point, traditional MAB algorithms will be inadequate.

The integration of MAB with deep learning aims to leverage the powerful feature extraction and pattern recognition capabilities of deep neural networks to solve these complex problems [8].

The intersection of reinforcement learning and MAB: MAB can be regarded as a simplified version of reinforcement learning, and the boundary between them is becoming increasingly blurred. Many studies utilize the framework of reinforcement learning to solve more complex MAB problems. For instance, when the selection of the arm affects the future context state, the issue of state transition needs to be considered.

6. Conclusion

This study, through review and empirical analysis, compared the core ideas and performance of three classic multi-armed bandit algorithms - UCB, ETC, and Thompson sampling. The case study conducted using the MovieLens 1M dataset visually verified the following main findings:

Thompson Sampling performed the best in terms of cumulative regret, with its curve growing the slowest and reaching the lowest final value. This confirms the superiority of its Bayesian approach in balancing exploration and exploitation, making it a powerful tool for handling sequential decision-making problems. In contrast, the UCB algorithm performed secondarily, with its logarithmic growth regret curve aligning with its strategy of continuous exploration, being a theoretically robust and widely applicable choice. The ETC algorithm, however, performed relatively poorly due to its lack of dynamic exploration capabilities.

In conclusion, this study not only systematically reviewed the theoretical basis of these algorithms, but also verified their advantages and disadvantages through actual data. We provided valuable insights for making decisions in uncertain environments, and emphasized that in future research, we should further explore directions such as contextual information, non-stationary environments, and algorithmic hybrid strategies.

References

[1] J. Zhao, "Comparison of Multi-Armed Bandit Algorithms in advertising recommendation Systems," Applied and Computational Engineering, vol. 83, no. 1, pp. 62–71, Oct. 2024, doi: 10.54254/2755-2721/83/2024glg0074.

[2] S. Zhang, "Optimizing data filtering in Multi-Armed Bandit Algorithms for reinforcement learning," ITM Web of Conferences, vol. 73, p. 01024, Jan. 2025, doi: 10.1051/itmconf/20257301024.

[3] H. Qi, F. Guo, and L. Zhu, "Thompson Sampling for Non-Stationary bandit Problems," Entropy, vol. 27, no. 1, p. 51, Jan. 2025, doi: 10.3390/e27010051.

[4] Y. Lei, "Comparative Evaluation of Mean Cumulative Regret in Multi-Armed Bandit Algorithms: ETC, UCB, Asymptotically Optimal UCB, and TS," ITM Web of Conferences, vol. 73, p. 01026, Jan. 2025, doi: 10.1051/itmconf/20257301026.

[5] "MovieLens 1M Dataset," GroupLens, Mar. 02, 2021. https://grouplens.org/datasets/movielens/1m/

[6] J.-M. Kim, "LLM-Guided Ensemble Learning for Contextual Bandits with Copula and Gaussian Process Models," Mathematics, vol. 13, no. 15, p. 2523, Aug. 2025, doi: 10.3390/math13152523.

[7] X. Zhan, "A review of Multi-Armed Bandit Algorithms in player modeling and game design," ITM Web of Conferences, vol. 73, p. 02016, Jan. 2025, doi: 10.1051/itmconf/20257302016.

[8] N. Y. Zhou, "The application of deep network and reinforcement learning for art design in graphical user interface wireframe generation," Automatic Control and Computer Sciences, vol. 59, no. 3, pp. 402–415, Jun. 2025, doi: 10.3103/s0146411625700567.