Comparison of Video Recommendation Effects of Etc, Ucb, and Thompson Sampling Algorithms on Short-Video Platforms

Shouchuan Li

Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, 999078, China

Abstract. This paper comprehensively compares the performance of three multi-armed bandit (MAB) algorithms, Epsilon-Then-Commit (ETC), upper confidence bound (UCB), and Thompson sampling (TS), for video recommendation in dynamic environments. Using real TikTok interaction data, their performance is evaluated in both static and dynamic scenarios, where user preferences change. Experimental results show that Thompson sampling performs best, while ETC can only re-adapt to the environment after the environment changes due to its fixed exploration strategy. UCB shows moderate adaptability and relies on the adjustment of confidence intervals. TS's Bayesian approach can naturally balance exploration and exploitation without manual parameter tuning, and TS can achieve faster convergence recovery than the other two algorithms. Although TS has high computational overhead and long running time, its robustness in dynamic scenarios proves its application value in practical recommendation systems. This study highlights the importance of adaptive exploration mechanisms in dealing with non-stationary user behaviours and provides a reference for deploying multi-armed bandit algorithms on large-scale platforms. Future research directions include combining contextual information and hybrid models with deep learning techniques.

1 Introduction

1.1 The Rise of Short Video Platforms and Challenges

The explosive growth of short video platforms in recent years has fundamentally changed the consumption pattern of digital content. Platforms such as TikTok and YouTube now have billions of monthly active users, with users spending an average of more than 90 minutes per day watching short videos. This unprecedented scale of content consumption presents both opportunities and challenges for recommendation systems. How to match an infinite pool of content with users' limited attention spans while maintaining user stickiness and satisfaction becomes a major issue.

1230019966@student.must.edu.mo

1.2 Limitations of Traditional Recommendation Methods

Traditional recommendation methods, including collaborative filtering and deep learning-based models, face significant limitations in this dynamic environment [1]. These methods rely primarily on historical interaction data, making them passive rather than active in nature. Three key flaws emerge in the short video environment: (1) the cold start problem for new content, where new videos have difficulty gaining initial appeal; (2) the rapidly evolving nature of user preferences that may change within a single browsing session; and (3) the need to adapt to immediate user feedback signals such as viewing time, likes, and skips in real time [2].

1.3 MAB Framework as a Solution

The MAB framework has been valued as a theoretically plausible approach to address these challenges. By explicitly modelling the trade-off between exploration and exploitation, MAB algorithms can dynamically balance between testing the possibility of new content and exploiting known preferences [3]. This feature makes them particularly suitable for short video recommendation scenes.

1.4 Algorithm Comparison and Evaluation Metrics

In this study, three basic MAB algorithms- ETC, UCB, and TS-are compared in the context of short video recommendation. Experimental procedures are designed to compare and analyse their performance in short video recommendation scenarios. Based on the experimental results, multiple evaluation dimensions are used to comprehensively evaluate the performance of the algorithms. In addition to traditional indicators such as click-through rate (CTR), user engagement in terms of average viewing time, feedback behaviours such as likes and reposts, and system responsiveness to changes in user preferences are also examined. The evaluation framework combines experimental results from real-world interaction datasets with theoretical analysis of the algorithm's regret bound and convergence characteristics.

1.5 Research Purpose

This study aims to find the optimal algorithm recommendation strategy in different situations based on the data results of the three algorithms running in different situations, combined with other off-site factors such as video length, video type, and user information.

2 Related Work

2.1 Short Video Recommendation Systems

Different from traditional recommendation scenarios, modern short video recommendation systems face unique challenges. The ultra-short content duration of typically 15-60 seconds creates an unprecedented decision density, and the system is required to make accurate recommendations within milliseconds after each interaction. Unlike movie or e-commerce recommendations, where user preferences evolve gradually, short video platforms must cope with rapid preference changes that may occur during a single viewing session. This time sensitivity is further complicated by the “swipe-next” interaction pattern, where each swipe represents an implicit rejection of the current recommendation.

Recent advances have focused on addressing these challenges through various approaches. Deep learning architectures, especially transformer-based models, have shown promise in capturing sequential patterns of user behaviour [4]. However, these approaches often struggle with the cold start problem when faced with emerging content and require a large amount of computation for real-time inference [5].

2.2 Application of MAB Algorithms

The MAB framework has emerged as a compelling solution to address the dynamic nature of short video recommendations. In the context of recommender systems, MAB algorithms have shown effectiveness in three key areas: content exploration, user preference learning, and real-time adaptation. The ETC algorithm, despite its simple implementation, has been widely adopted in industrial settings due to its straightforward parameter tuning and predictable behaviour. However, its fixed exploration rate often leads to poor performance in environments where user preferences change rapidly. More sophisticated methods, such as UCB and Thompson sampling, have gained academic attention due to their theoretical guarantees and empirical performance. The UCB algorithm has shown strong performance in scenarios requiring fast convergence, thanks to its optimistic principle in the face of uncertainty. Variants such as Linear Upper Confidence Bound (LinUCB) have extended this approach to incorporate contextual information, demonstrating its value in personalized recommendation tasks [6]. Thompson Sampling, based on Bayesian theory, has shown effectiveness in balancing exploration and exploitation, often outperforming other methods in A/B testing scenarios, despite its higher computational requirements.

Recent research has focused on hybrid approaches that combine the strengths of different MAB algorithms. Some studies have proposed adaptive ε policies that automatically adjust the exploration rate based on system performance [2]. Others have studied hierarchical bandit structures to handle the large action space unique to short video platforms [7]. Integration of bandit algorithms with deep learning models has shown promise, although it introduces additional complexity in model training and deployment.

3 Methodology

3.1 Experimental Environment Setup

The experimental scenario environment can be divided into two categories: static and dynamic. By determining whether the user's recent interest is stable based on the user's historical behaviour, if the recent behaviour does not change much from the historical behaviour, it is approximately treated as a static environment. In this static environment, the MAB algorithm is used to establish a recommendation model. However, it is inevitable that new videos will suddenly become popular, causing user interests to change suddenly. At this time, the user's interest should be analysed as a dynamic environment, and the recommendation model should be rebuilt based on the algorithm theory of a dynamic environment.

3.2 Problem Modelling

This study models the short video recommendation task as a random multi-armed bandit problem with K arms. Each video type or video set is regarded as an arm. In each T period, the recommendation system selects an arm $A \in \{1, \dots\dots, K\}$ and observes the reward $R \in [0,1]$ representing user engagement. Here, R is defined as R = (1*video likes + 2*video shares + 3*video comments)/ (video views). Each arm i has an unknown expected reward $\mu_i = E[R|A = i]$, and the reward of the optimal arm is $\mu^* = max_i\mu_i$ [8]. The goal of the study is to minimize the cumulative regret.

$$R(T) = T\mu^* - E[\sum_{t=1}^{T} r_t] \tag{1}$$

3.3 Analysis of Three Classic MAB Algorithms

3.3.1 ETC

The ETC algorithm runs in two phases. In the exploration phase (the first $\tau = \varepsilon T$ rounds), the arm is uniformly randomly selected. Suppose the number of times arm i is selected during the exploration stage is $n_i(\tau)$. Record the rewards obtained by each arm i and calculate its empirical average reward:

$$\hat{\mu}_i(\tau) = \frac{1}{n_i(\tau)} \sum_{t=1}^{\tau} r_t \cdot I\{a_t = i\} \tag{2}$$

In the utilization phase (the remaining $T - \tau$ rounds), the empirical best arm $\hat{\imath} = argmax_i\hat{\mu}_i(\tau)$ is fixedly selected. The ETC algorithm has two key parameters, the exploration phase ratio $\varepsilon \in (0,1)$ and the total time horizon T, which is known in advance [9]. Its theoretical optimal regret bound is when $\varepsilon = \sqrt[3]{\frac{K}{T}}$.

$$R(T) = O\left(K^{\frac{1}{3}}T^{\frac{2}{3}}\right) \tag{3}$$

3.3.2 UCB

Like the ETC algorithm, the UCB algorithm also has two key parameters, the empirical average reward $\mu_i(t)$ of arm i at time t, and the number of times arm i is selected at time t which is $n_i(t)$. The UCB algorithm adheres to the following selection rule:

$$a_t = argmax_i \left[\mu_i(t-1) + c\sqrt{\left(\frac{2l\,n(t)}{n_i(t-1)}\right)}\right] \tag{4}$$

The UCB algorithm has three theoretical properties: optimal logarithmic regret, suboptimal gap $\Delta i = \mu^* - \mu_i$, and the ability to automatically adapt to the reward distribution [10].

$$R(T) \le \sum_{\Delta i>0} \frac{8lnT}{\Delta i} + \left(1 + \frac{\pi^2}{3}\right)\Delta i \tag{5}$$

3.3.3 TS

In the Thompson Sampling algorithm, the standard Beta-Bernoulli model assumes that the reward is binary ($r_t \in \{0,1\}$), and thus is directly applicable to the Bernoulli distribution; its prior is $\mu_i \sim \mathrm{Beta}(\alpha_i, \beta_i)$. However, the reward in this experiment is a continuous value ($r_t \in$

$[\mathrm{a,b}]$), and it needs to be normalized to the interval [0,1]. Suppose the reward is $\mathrm{r_t} \in [\mathrm{r_{min}}, \mathrm{r_{max}}]$. It can be mapped to [0,1] through a linear transformation:

$$\widetilde{r_t} = \frac{r_t - r_{min}}{r_{max} - r_{min}} \tag{6}$$

Subsequently, the weighted update is adopted, and the parameters are updated directly in proportion to the normalized rewards $(\alpha_i, \beta_i) \leftarrow (\alpha_i + \widetilde{r_t}, \beta_i + 1 - \widetilde{r_t})$. Next, sample each arm in each round $\theta_i \sim Beta(\alpha_i, \beta_i)$. Finally, select based on $\alpha_t = argmax_i \theta_i$. Thompson sampling theory guarantees asymptotically optimal regret and has a strong dynamic adaptation environment [11].

$$R(T) = O\left(\sum_{i \neq i^*} \frac{lnT}{\Delta i}\right) \tag{7}$$

3.4 Theoretical Comparisons

Table 1 Theoretical comparison of the three algorithms

Algorithms	ETC	UCB	TS
Worst case regret	$O\left(K^{\frac{1}{3}}T^{\frac{2}{3}}\right)$	$O(\sqrt{KTlnT})$	$O(\sqrt{KTlnT})$
Computational complexity	$O(1)$	$O(K)$	$O(K)$
Parameter sensitivity	High	Medium	Low
Adaptability to non-static environments	Poor (fixed exploration)	Medium (decayed exploration)	Excellent (natural forgetting)

In Table 1, the comparisons of three MAB algorithms in aspects of worst case regret, computational complexity, parameter sensitivity, and adaptability are presented.

4 Experiment and Result Analysis

4.1 Dataset Selection and Processing

From as many as 19,382 videos, 1,000 videos were randomly selected for data pre-processing. These 1,000 videos were used as the 1,000 arms of the experiment. This experiment divided these randomly extracted videos into five categories according to their type characteristics: science and technology, social hot spots, history and humanities, games and animation, and others (such as natural scenery types with a small number of videos). Extract the attribute values in the extracted dataset: video type, video playback volume, video likes, video forwarding volume, and video comments. Weight the extracted attribute values of each video and calculate the reward.

4.2 Experimental Design Ideas of Three MAB Algorithms

4.2.1 ETC

In the design of the ETC algorithm, this experiment fixed $\varepsilon = 0.1$. After multiple experiments, we evaluated the performance of different ε values in exploration and utilization. Finally, ε was determined to be 0.1, which can minimize regret and form a sharp contrast with the other two algorithms. Set T as the total number of rounds, n as the number of arms, and the number of rounds explored for each arm as $\frac{\varepsilon T}{n}$. This ensures that each arm can be explored and increases the probability of selecting the optimal arm at the end of the exploration phase and utilizing it.

4.2.2 UCB

For the UCB algorithm, this experiment sets the confidence coefficient c in the range of 1~4 and adjusts it according to recent performance. Increase c when the reward decreases to enhance exploration and decrease c when the reward is stable to enhance utilization, ensuring that implicit exploration can be automatically achieved through confidence interval expansion in a dynamic environment, and finally converge to the logarithmic regret bound.

4.2.3 TS

The core idea of Thompson Sampling is the Bayesian probability matching method. This experiment sets up a key mechanism in its probability sampling, sampling from the β distribution of each arm and selecting the maximum value, adjusting the normalized reward to the $[0,1]$ interval, and then updating the α and β parameters of the selected arm. The theoretical advantage of doing so is that it can naturally achieve a balance between exploration and utilization and is not afraid of the influence of the dynamic environment.

4.3 Environment Establishment

4.3.1 Static environment

The reward distribution is fixed. This experiment assumes that the user's preferences remain unchanged and the popularity of the external video type remains unchanged. The purpose of establishing this environment is to verify whether the three MAB algorithms can maintain theoretical performance in short video recommendations.

4.3.2 Dynamic Environment

In the current short video network environment, the iteration speed of popular video types is very fast, and a certain type of video will suddenly become popular. Therefore, it is necessary to establish a dynamic environment on the premise that the three MAB algorithms can guarantee theoretical performance. This experiment divides the entire recommendation process into three sections, and designs two inflection points at T/3 and 2T/3, increasing the rewards for science and technology videos and animal videos by 50% respectively. This requires the three algorithms to re-explore each time they pass a turning point. However, to fit the actual situation, this experiment chooses to retain 50% of the arms that have been explored and have high reward values each time the environment changes. This design can also reduce computational costs.

4.3.3 Experimental Design

Total rounds $\mathrm{T} = 100{,}000$. Repeat the experiment 100 times and take the average. Draw performance comparison curves of the three algorithms in static and dynamic environments.

4.4 Evaluation Metrics

To better evaluate the performance of the three algorithms, this experiment set up three evaluation criteria. The first one is Cumulative regret:

$$R(T) = \sum_{t=1}^{T} (\mu^* - \mu_t) \tag{8}$$

Here, T represents the total number of rounds of the experiment, μ^* is the expected reward of the optimal arm in the new environment after the change. μ_t is the expected reward of the arm selected by the algorithm at time t. It can be seen from the formula that the smaller the cumulative regret value of this algorithm is, the better its performance will be. The second one is Average reward:

$$\bar{r}(T) = \frac{1}{T} \sum_{t=1}^{T} r_t \tag{9}$$

In this formula, r_t represents the reward observed at time t. This evaluation criterion reflects the quality of the recommendation. The last assessment criterion is recovery speed. It is to compare which of the three algorithms requires fewer rounds to reach the stable optimal state again after the environmental change. In this experiment, the optimal reward after the environmental change is set to μ_{new}. When the average reward of the algorithm is greater than $95\%\mu_{new}$, it is regarded as having recovered. This evaluation criterion reflects the adaptability of the algorithm to the dynamic environment.

4.5 The Result of The Experiment

4.5.1 Results in Static Environment

Graph analysis of three algorithms in a static environment (Figure 1). The three algorithms present theoretical comparison results. TS performs best and can achieve fast convergence. UCB is second. To select the optimal arm during utilization, ETC wastes too many resources in exploration, and the performance of the ETC algorithm will deteriorate with the increase of implementation rounds and the increase of scale. However, it is worth noting that when the total rounds are less than 10,000, the ETC algorithm will slowly recover its disadvantage, and with its own advantages of simple calculation and short time consumption, it may even be better than the TS algorithm.

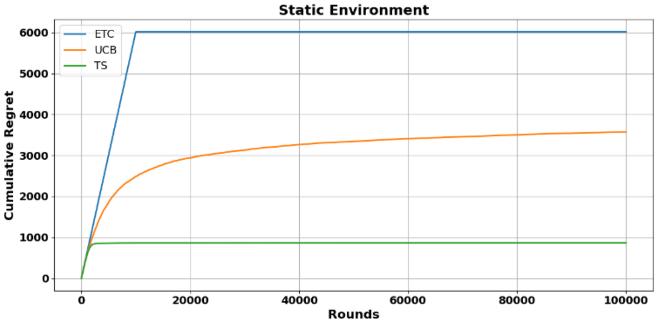

Figure 1 Results in a static environment (Photo credit: Original)

4.5.2 Results in Dynamic Environment

Graph analysis of three algorithms in a dynamic environment (Figure 2). All algorithms have obvious inflection points at the change points of 33,333 and 66,666 rounds. The more inflection points, the greater the final regret gap between the three algorithms. In a dynamic environment, the TS algorithm has the best performance in terms of total regret, convergence, and recovery ability after each re-exploration. It shows better adaptability. Although the UCB algorithm performs well in the utilization stage and recovery ability after convergence, it generally performs poorly in the exploration stage. However, the ETC algorithm must completely re-explore each time it passes an inflection point because of the fixed value of the ε parameter, which makes it difficult to cope with changing dynamic situations. Due to the mechanism characteristics of retaining 50% of the historical exploration data set in this experiment, the algorithm will converge faster than the previous time after each inflection point (except for the ETC algorithm). It can also be seen that the average regret growth rate of the three algorithms is decreasing at each stage.

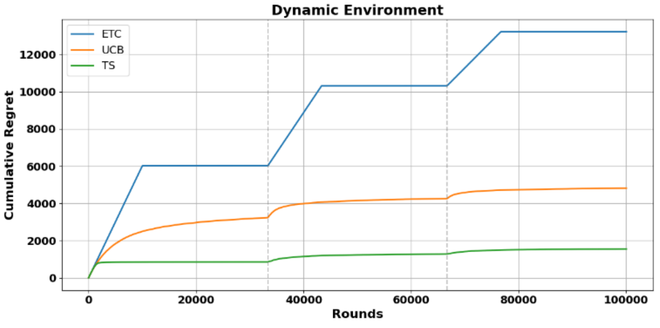

Figure 2 Results in a dynamic environment (Photo credit: Original)

4.6 Summary

After comparing the data and images of the three algorithms, it is not difficult to see that TS performs best in the cold start scenario because its probabilistic sampling mechanism can quickly explore high-quality new arms. In a dynamic environment, UCB and TS are more suitable for short video recommendation scenarios where user interests change rapidly, while ETC has the highest computational efficiency. The single-round calculation time is 0.12ms, which is better than UCB's 0.25ms and TS's 0.38s. So, it is suitable for systems with strict real-time requirements.

This experiment only randomly selected 1,000 videos from a large data set, which is random. Although 1,000 arms are convenient for calculation, there may be some errors in the results due to the insufficient scale of the experiment and the insufficient number of arms. The video classification is not detailed enough, which may have a slight impact on the calculation of accumulated regret when the environment changes. However, this experiment is a simulation experiment, the purpose of which is to compare the video recommendation effects of ETC, UCB, and TS algorithms on short video platforms. A small data error will not affect the experimental conclusion.

5 Conclusion

This study systematically evaluates the performance of three MAB algorithms—ETC, UCB, and TS—in the short video recommendation scenario, focusing on static and dynamic environments. The experimental results demonstrate that in a dynamic environment, Thompson sampling consistently outperforms the other two algorithms in terms of cumulative regret value, adaptation speed, and robustness to environmental changes. Thompson sampling naturally balances exploration and exploitation through Bayesian posterior sampling without manual parameter tuning, whereas ETC and UCB require careful calibration of their exploration parameters ε and c. Additionally, the probabilistic framework of TS enables quick adaptation to sudden changes in user preferences, allowing it to recover optimal performance rapidly after transitions. In contrast, ETC requires a significantly longer adjustment period, making TS particularly suitable for real-world platforms where user behaviour is unpredictable.

However, this study has certain limitations, such as not incorporating user and video features, which may slightly impact reward evaluation. Extending the TS algorithm to contextual bandits, such as LinUCB or Neural Thompson Sampling (NeuralTS), could further enhance personalization. Future research should explore testing on larger datasets, including billions of user logs in industrial deployments, where integrating deep learning architectures like Transformer-based recommendation systems could be key to optimizing recommendation efficiency and accuracy. In summary, Thompson sampling performed exceptionally well in this experiment, proving to be the most effective MAB algorithm for dynamic recommendation tasks due to its adaptability and time-sensitive exploration. Future work should focus on developing hybrid models that combine TS's strengths with the scalability of modern deep learning methods to further advance recommendation systems.

References

1. Zhang, E., Ma, W., Zhang, J., et al.: ‘A Service Recommendation System Based on Dynamic User Groups and Reinforcement Learning’. Electronics, 2023, 12, (24), 5034.
2. Fenglin, Lu.: 'Adaptive Recommendation Sym Strategies: An Exploration of Online Machine Learning Algorithms', EMITI, 2024, pp. 447-451
3. Elena, G., Milos, K., Eugene, I.: 'Survey of multiarmed bandit algorithms applied to recommendation systems', Int. J. Open Inf. Technol., 2021, 9, (4), pp. 12-27
4. Singh, S., Mahmood, A.: 'The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures', IEEE Access, 2021, 9, pp. 68675-68702

5. Jin, H.: 'Optimizing Short Video Recommendation Systems: Addressing Cold-Start and Diversity Challenges through Advanced Algorithms', Appl. Comput. Eng., 2024, 83, pp. 72-80
6. Song, Y.: 'Truncated LinUCB for Stochastic Linear Bandits'. PhD thesis, Queen's University, 2022
7. Guo, H., Naeff, R., Nikulkov, A., et al.: 'Evaluating Online Bandit Exploration in Large-Scale Recommender System'. PhD thesis, Stanford University, 2023
8. Lattimore, T., Szepesvári, C.: 'Bandit algorithms' (Cambridge University Press, 2020)
9. De Ath, G., Everson, R.M., Fieldsend, J.E.: 'Asynchronous ε-Greedy Bayesian Optimisation'. Proc. 37th Conf. Uncertainty in Artificial Intelligence, Exeter, United Kingdom, June 2021, pp. 578-588
10. Du, M.: 'Applications and Advances of UCB Algorithms in Dynamic and Contaminated Environments', Appl. Comput. Eng., 2024, 83, pp. 155-165
11. Agrawal, S., Goyal, N.: 'Analysis of Thompson sampling'. Proc. 25th Annual Conf. Learning Theory, Edinburgh, UK, June 2012, pp. 39.1-39.26