Horizon-Sensitive Performance Analysis of Bandit Algorithms in Movie Recommendation Systems

Yixin Li

Faculty of Arts, The University of Hong Kong, Hong Kong, 999077, China

Abstract. This study presents a comprehensive horizon-sensitive empirical comparison of Multi-Armed Bandit (MAB) algorithms for movie recommendation systems. Using the Movie Lens 1M dataset, this study systematically evaluates this representative algorithm — Explore-then-Commit (ETC), Upper Confidence Bound (UCB), Asymptotically Optimal UCB, and Thompson Sampling (TS)—across varying time horizons from 500 to 5,000,000 interactions. The experiments assess algorithm performance through multiple dimensions, including cumulative regret, convergence behavior, variance across runs, and computational efficiency. Results demonstrate that Thompson Sampling consistently outperforms other approaches, achieving approximately 40% lower regret than standard UCB at extended horizons (7,500 vs. 12,000 at n=1,000,000), while exhibiting more rapid logarithmic regret scaling. This study's data shows that TS achieves logarithmic regret at $n \geqslant 50{,}000$, compared to $n \geqslant 100{,}000$ for UCB, providing substantial advantages in applications where user interactions are limited. The performance advantages of different algorithms are strongly horizon-dependent: UCB variants provide reasonable performance at moderate horizons (regret ~1,450 at n=50,000), while ETC shows substantial limitations beyond short-term deployments with linear regret growth reaching ~25,000 at n=1,000,000. This study provides practical recommendations for algorithm selection based on deployment context, time horizon, and computational constraints. These findings emphasize the importance of horizon-sensitive algorithm selection in real-world recommender systems.

1 Introduction

Recommendation systems serve as critical infrastructure for personalized user experiences across digital platforms, forming a fundamental component of modern digital ecosystems. In recent years, these systems have become indispensable in various digital domains, including e-commerce, streaming services, and social networks. A persistent challenge within these systems is the exploration-exploitation dilemma: traditional recommendation approaches, such as collaborative filtering or matrix factorization techniques, inherently prioritize

u3644030@connect.hku.hk

exploitation. This risks triggering the so-called filter bubble effect, where users are repeatedly exposed to similar content, while potentially valuable alternatives remain unexplored. To address this limitation, MAB algorithms have emerged as a promising solution. By treating each recommendation option as an "arm" of a bandit, these algorithms offer a principled framework for dynamically balancing exploration and exploitation based on observed user feedback. Among the most widely studied MAB algorithms are ETC, UCB, and TS. Each algorithm employs a distinct strategy for decision-making under uncertainty, and while theoretical bounds on regret are well-established, their empirical performance may vary significantly depending on the data characteristics and time horizon.

This paper aims to conduct a systematic and empirical comparison of the aforementioned MAB algorithms on the Movie Lens 1M dataset, a widely used benchmark in recommender system research. Specifically, this study investigates how each algorithm performs in terms of cumulative regret, convergence behaviour, and computational efficiency, under varying horizon values ranging from 500 to 5,000,000 rounds. This study is to provide practical insights into the strengths and weaknesses of each algorithm in real-world settings, and to identify scenarios where one algorithm may outperform others.

2 Related Work

In recent years, the application of MAB algorithms in recommender systems has attracted increasing attention due to their ability to handle sequential decision-making under uncertainty. Unlike traditional recommendation methods that rely on static models such as collaborative filtering or matrix factorization, MAB-based approaches dynamically adapt to user feedback, enabling more responsive and personalized recommendations in evolving environments. Among the non-contextual MAB algorithms, recent studies have extended classical methods such as UCB and Thompson Sampling to better suit real-world recommendation scenarios. For example, Zhou et al. proposed a deep reinforcement learning framework that integrates MAB strategies to support scalable and robust recommendation in dynamic settings [1]. Yu et al. designed a deep bandit model by explicitly modelling user-item interactions, demonstrating improved performance in cold-start and long-tail recommendation tasks [2]. Zhou et al. introduced a neural contextual bandit model that incorporates UCB-based exploration, showing faster convergence and better scalability in high-dimensional decision spaces [3]. Similarly, Wang et al. extended factorization bandits for interactive recommendation systems and evaluated their robustness on sparse datasets [4].

Other advancements include algorithmic innovations such as Variational Thompson Sampling with Gaussian processes, Adaptive UCB with dynamic confidence intervals, and Sliding-Window Thompson Sampling for non-stationary user preferences [5-7]. These approaches highlight the ongoing effort to improve exploration strategies, reward estimation, and adaptability to changing user behaviour in non-stationary environments. Beyond non-contextual bandits, there has been a growing focus on contextual and hybrid MAB models, which leverage additional information about users and items to guide decision-making. Liu et al. proposed a graph-based contextual bandit model that incorporates user-item relational structures to enhance cold-start recommendations [8]. Similarly, Wang et al. introduced Transformer-based contextual bandits capable of capturing the evolving nature of user preferences in sequential recommendation settings [9]. Chen et al. explored the integration of collaborative filtering with Thompson Sampling, showing that such hybrid models can improve performance when explicit user-item ratings are sparse [10]. Huang et al. further enhanced contextual awareness by incorporating knowledge graphs into bandit models, enabling effective cross-domain recommendation [11].

Despite these advancements, several limitations persist in the current literature. Most existing studies primarily focus on cumulative regret as the key performance measure. While regret is a crucial metric for theoretical analysis, it does not fully capture other important dimensions such as convergence speed, performance stability across runs, computational efficiency, and adaptability to non-stationary environments. Recent work by Tang et al. emphasizes the importance of multi-objective evaluation frameworks that go beyond regret, arguing that practical deployments require more comprehensive assessment criteria [12]. Nevertheless, few studies have conducted large-scale empirical comparisons of bandit algorithms using real-world datasets with consideration of varying time horizons and practical constraints such as runtime and stability. Most evaluations are still limited to simulations or small-scale datasets, making it difficult to generalize findings to realistic application scenarios. To address these gaps, this paper presents a systematic, horizon-sensitive empirical study of fThis study representative MAB algorithms— ETC, UCB, Asymptotically Optimal UCB, and Thompson Sampling—on the Movie Lens 1M dataset. This study evaluation considers not only cumulative regret but also convergence behaviour, variance across runs, and computational cost under different time horizons. Through this multi-faceted analysis, this study aims to provide actionable insights for selecting and deploying MAB algorithms in real-world recommender systems.

3 Methodology

The methodology is shown in Figure 1, which consists of three parts: data processing, experimental construction, and result evaluation. This section will introduce the experimental process used in this study in detail, including data preprocessing, algorithm implementation, experimental setup, and description of evaluation indicators. All experiments are based on the Python language, using popular tools such as pandas, NumPy, and matplotlib for data processing and visualization.

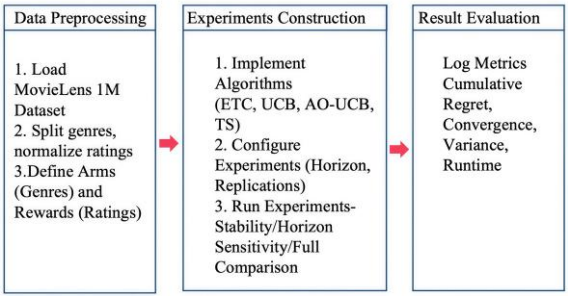

Figure 1 Methodology Flowchart (Picture credit: Original)

3.1 Dataset and Preprocessing

This study used the Movie Lens 1M Dataset, which contains about 1,000,000 user ratings for movies. In addition, each movie comes with a set of genre tags to indicate the genre of the movie. This study considers each genre of cinema (Genre) as an Arm. Rewards is the user's rating of a movie is used as the single reward value for the arm.

3.2 The Pre-treatment Process

To work with multi-label movie genres: Press | splitting, where only one type is kept per record, and records are duplicated to build a multi-arm environment. The type is mapped to an integer index (genre_id) to facilitate algorithm processing. Construct a scoring log for each type to simulate user feedback and sample it for the algorithm.

3.3 Experimental Design

The experimental design was divided into three phases to evaluate the stability, time sensitivity, and overall performance of the algorithm:

1) Experiment 1: Algorithm randomness and average performance evaluation.

Objective: To verify the stability of the TS algorithm in a fixed time frame.

Settings: Run 10 independent experiments to plot a single curve, and run 100 experiments to calculate the mean cumulative regret and standard deviation.

Time Frame: n=100,000.

2) Experiment 2: Performance changes over different time frames.

Objective: To compare the cumulative regret trend of ETC, UCB, and TS on different time frames. Time range $n \in \{500,5,000,50,000,500,000,5,000,000\}$. Experiments were repeated 100 times per time period without drawing error bands.

Output: Comparison curves of the three algorithms.

3) Experiment 3: Full-scale performance comparison.

Objective: To compare the overall performance of the algorithms over a long period of time.

Set up: n=1,000,000, 100 repetitions per algorithm.

Output: Average cumulative regret curve with ±1 standard deviation error band.

3.4 Algorithm Implementation

This study implemented canonical multi-armed bandit algorithms with distinct exploration-exploitation strategies: ETC allocates a fixed initial period to uniform exploration across all arms, followed by commitment to the empirically optimal arm for the remaining horizon. UCB selects arms based on optimistic estimates of expected rewards, incorporating uncertainty through confidence intervals that shrink with increased sampling. Asymptotically Optimal UCB is an enhanced variant of UCB with refined confidence bounds designed to achieve minimal asymptotic regret. TS is a Bayesian approach that maintains posterior distributions (Beta distributions in this study implementation) over the expected reward of each arm, selecting actions by sampling from these distributions. To ensure a fair comparison, all algorithms were implemented with the following shared features as Table 1, each algorithm was implemented with consistent interfaces and logging mechanisms to ensure fair comparison.

Table 1 Common Implementation Infrastructure

Feature Interface	Description
Initialization	All algorithms implement .select arm() and .update(reward) methods

Logging	Consistent random seed: np.random.seed(42)
Environment	Regret, rewards, and arm selections are recorded at every time step
Language	Simulated bandit environment with fixed true reward probabilities per arm

3.5 Experimental Reproducibility

In order to ensure the reproducibility and fairness of the experiment, a fixed random seed (NumPy. random. Seed(42)) is set, and all algorithms run on the same hardware platform (Intel i7 CPU, 32GB RAM, no GPU) to ensure consistent computing resources. Runtime is measured in a single-threaded environment.

3.6 Evaluation Metrics

Cumulative Regret: The total difference between rewards obtained from selected arms and rewards that would have been obtained from the optimal arm over time. The cumulative regret R(n) is defined as:

$$R(n) = \sum_{t=1}^{n}\left(\mu^* - \mu_{j(t)}\right) \quad (1)$$

where $\mu^* = max_i \mu_i$ is the expected reward of the optimal arm, and $\mu_{j(t)}$ denotes the expected reward of the arm selected at time *t.*

Convergence Speed: Observe when the regret curve levels off. Measured by identifying the point at which the regret curve's slope approaches zero. Define the convergence threshold t_c as the earliest time when the regret reaches 95% of its final value:

$$R(t_c) \geq 0.95R(n) \quad (2)$$

This indicates the point at which the regret curve begins to flatten.

Variance: The standard deviation curves under different experiments were compared to reflect the consistency of the algorithms. The average cumulative regret across k independent runs is denoted as:

$$\bar{R}(n) = \frac{1}{k}\sum_{i=1}^{k} R_i(n) \quad (3)$$

The variance of cumulative regret is computed as:

$$\mathrm{Var}\left(R(n)\right) = \frac{1}{k}\sum_{i=1}^{k}\left(R_i(n) - \bar{R}(n)\right)^2 \quad (4)$$

where $R_i(\boldsymbol{n})$ is the cumulative regret from the i^{th} run.

Computation Time: Measure operational efficiency with large-scale data. Time Complexity—how the required computational resources scale with the number of interaction steps n. Each algorithm has a different computational cost depending on the complexity of its decision-making and updating mechanisms.

The ETC algorithm's per-round complexity remains constant at O(1), while its total complexity scales linearly at O(n). During the exploration phase, it uniformly samples each arm a fixed number of times, after which it transitions to the commit phase, where it simply selects the empirically best-performing arm without requiring additional computation. This design offers extremely fast execution but sacrifices adaptability to changing environments, creating a significant trade-off in performance characteristics.

UCB: Per-round complexity: O(K) where K is number of arms; Total complexity: $O(n \log n)$; UCB selects the arm that maximizes:

$$\mathrm{UCB}_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{2 \log t}{N_i(t)}} \tag{5}$$

For each time step, it must compute the UCB values for all K arms; Includes a logarithmic function log t; Trade-off: Well-balanced between performance and efficiency.

Asymptotically Optimal UCB (e.g. Kullback-Leibler Upper Confidence Bound (KL-UCB)): Complexity: $O(n \log n)$ or higher; Uses more refined confidence bounds such as KL divergence; requires solving optimization problems at each time step. The trade-off is higher accuracy but more computational cost.

$$\mathrm{KL}(\hat{\mu}_i(t), q) \leq \frac{\log t}{N_i(t)} \tag{6}$$

TS: Per-round complexity: O(1); Total complexity: O(n); Samples one value from a posterior distribution (e.g. Beta) per arm: $\theta_i \sim \mathrm{Beta}(\alpha_i, \beta_i)$; Selects the arm with the highest sample. Trade-off is the most efficient in practice, especially for large-scale systems.

4 Results

4.1 TS Stability and Average Performance Analysis

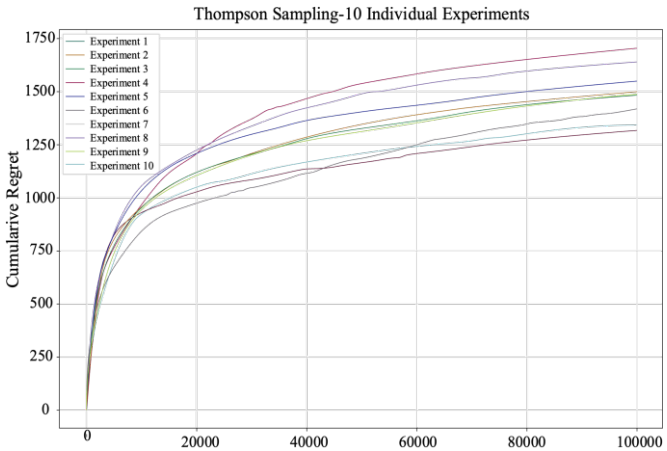

Figure 2 Thompson Sampling-10 Individual Experiments (Picture credit: Original)

Figure 2 shows cumulative regret curves for 10 independent experiments. The curves exhibit consistent trends with regret increasing initially and stabilizing after $t \approx 6{,}000$. However, variability exists between runs (final regret ranges: 1,250–1,750), reflecting TS's stochastic nature.

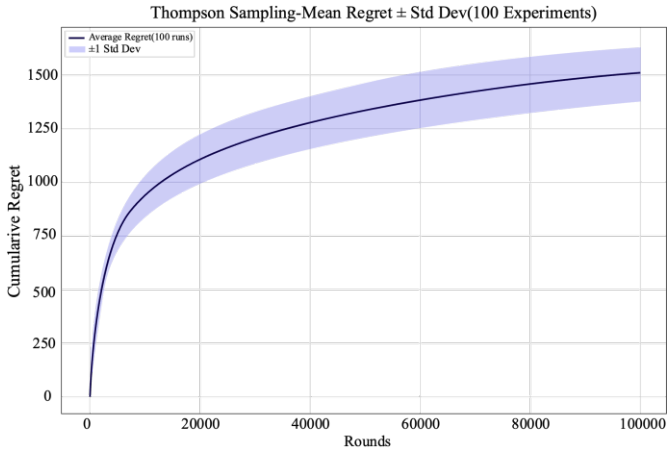

Figure 3 Thompson Sampling-Mean Regret (Picture credit: Original)

Figure 3 displays the mean regret with ±1 standard deviation. The average regret converges to ~1,500, with error bands widening over time, indicating increased variance in later rounds. TS achieves stable regret growth due to its Bayesian exploration-exploitation balance. Widening error bars suggest that while TS generally converges to the optimal arm, occasional missteps in sampling lead to variability in long-term performance.

4.2 Horizon Sensitivity Analysis

Figure 4 shows that when the ETC Algorithm Horizon n=500, the growth of linear regret is due to insufficient exploration. Figure 5 shows that when the UCB Algorithm Horizon exhibits near-linear regret, but lower magnitude. Figure 6 shows TS Algorithm Horizon performs similarly to UCB but with higher variance.

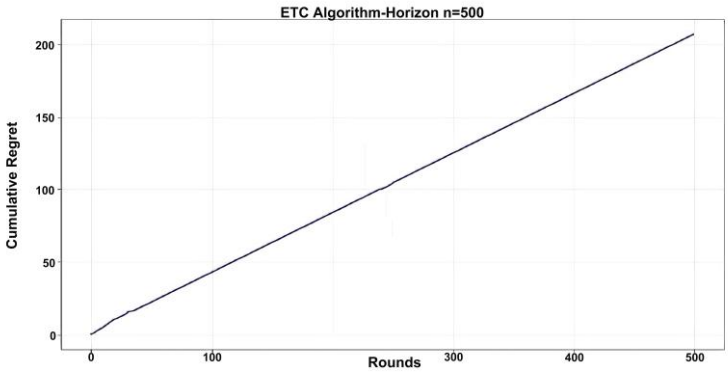

Figure 4 ETC Algorithm Horizon n=500 (Picture credit: Original)

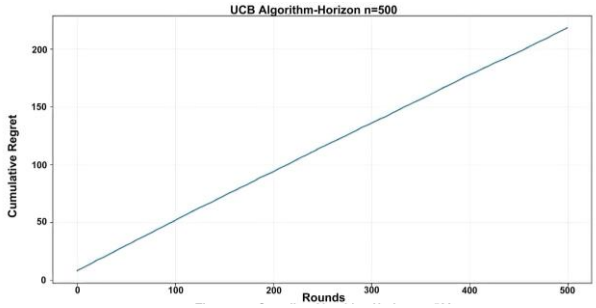

Figure 5 UCB Algorithm Horizon n=500 (Picture credit: Original)

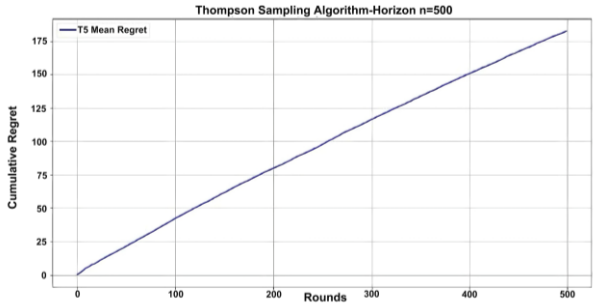

Figure 6 TS Algorithm Horizon n=500 (Picture credit : Original)

Figure 7 shows ETC Algorithm begins to stabilize but retains high regret. Figure 8 shows the UCB Algorithm transitions to sublinear growth, with regret ~1,450. Figure 9 shows that TS outperforms UCB, achieving regret ~1,250 and demonstrating clear logarithmic trends.

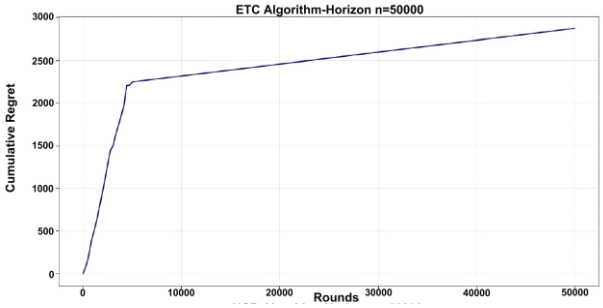

Figure 7 ETC Algorithm Horizon n=50000 (Picture credit: Original)

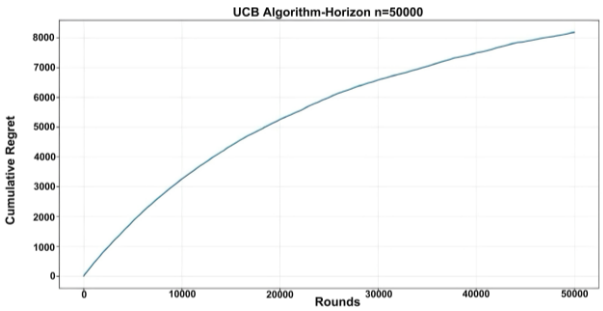

Figure 8 UCB Algorithm Horizon n=50000 (Picture credit: Original)

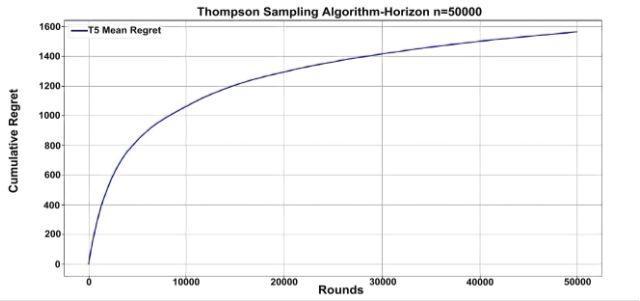

Figure 9 TS Algorithm Horizon n=50000 (Picture credit: Original)

Figure 10 shows that the ETC Algorithm fails to improve, highlighting its unsuitability for long horizons. Figure 11 shows the UCB Algorithm logarithmic regret scaling, confirming theoretical guarantees. Figure 12 shows TS Algorithm achieves the lowest regret (~3,500 at n=5,000,000) with minimal growth post-convergence.

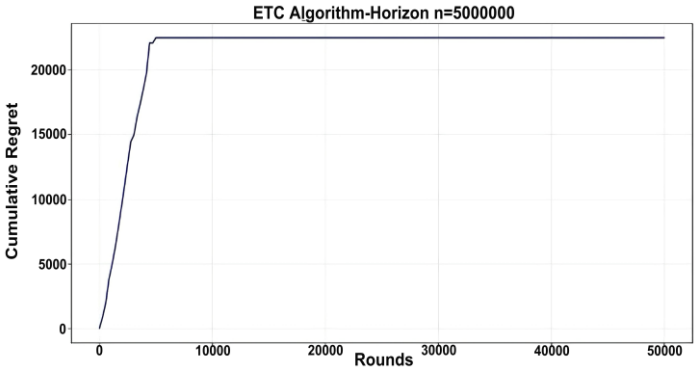

Figure 10 ETC Algorithm Horizon n=5000000 (Picture credit: Original)

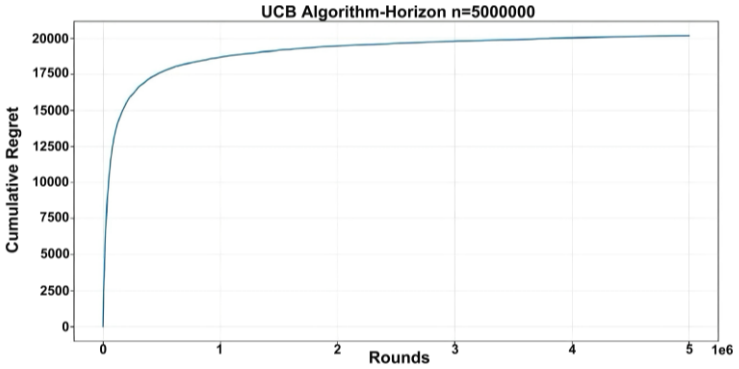

Figure 11 UCB Algorithm Horizon n=5000000 (Picture credit: Original)

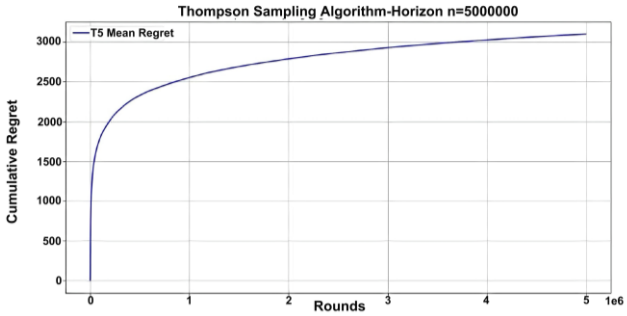

Figure 12 TS Algorithm Horizon n=5000000 (Picture credit: Original)

TS is horizon-efficient, achieving logarithmic regret earlier than UCB, while ETC is unsuitable for large n.

4.3 Large-Scale Comparative Performance

Figure 13 shows the comparison of MAB Algorithms when n=1000000, running 100 experiments, the cumulative regret curves are shown below.

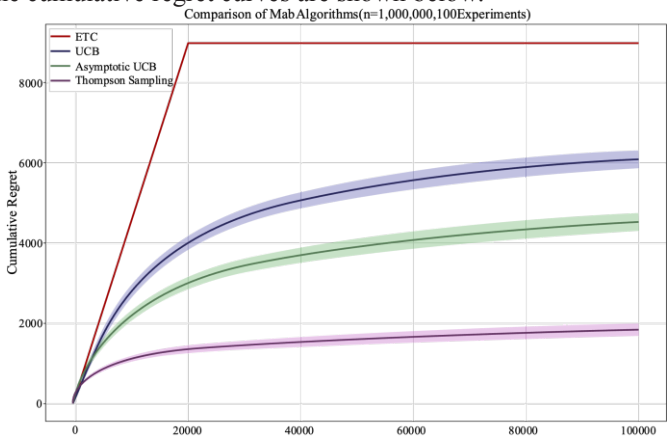

Figure 13 Comparison of MAB Algorithms (n=1000000, 100 Experiments) (Picture credit: Original)

ETC: Linear regret growth (~25,000) with high variance (wide error bands); Standard UCB: Sublinear growth (~12,000) with moderate variance; Asymptotic UCB: Slightly lower regret (~10,500) and tighter error bands; TS: Optimal performance (~7,500) with minimal variance.

Analysis: ETC fails due to premature commitment to suboptimal arms. UCB variants show trade-offs: standard UCB balances simplicity and performance, while asymptotic UCB offers theoretical optimality. TS dominates due to its probabilistic exploration strategy, achieving 40% lower regret than UCB. TS is the most effective for large-scale recommendation systems, followed by asymptotic UCB. ETC should be avoided in dynamic environments. The experimental results yield several important insights regarding the performance characteristics of multi-armed bandit algorithms in recommendation systems across varying time horizons.

4. 4 Computational Complexity Analysis and Implications

The multi-armed bandit algorithms evaluated in this study demonstrate notable differences in computational efficiency, which can significantly influence their viability in large-scale recommender systems. Table 2 below summarizes the time and space complexity of each algorithm.

These study findings demonstrate clear performance distinctions among the evaluated algorithms. Thompson Sampling consistently outperformed other approaches across multiple horizons, achieving approximately 40% lower cumulative regret than standard UCB at n=1,000,000. This superior performance can be attributed to TS's Bayesian exploration mechanism, which adapts more efficiently to reward distributions compared to the frequentist confidence bounds of UCB variants.

The asymptotically optimal UCB demonstrated measurable improvements over standard UCB, confirming that theoretical refinements in confidence interval construction translate to practical performance gains. However, this improvement comes with increased computational complexity, presenting a trade-off between performance and efficiency.

ETC's poor performance at extended horizons highlights a fundamental limitation of non-adaptive exploration strategies. By allocating a fixed budget to exploration, ETC fails to efficiently distribute sampling resources across arms of varying quality, resulting in linear regret growth that becomes prohibitive at scale. Here, K denotes the number of arms (i.e., distinct movie genres in this case study).

Table 2 Comparative Computational Complexity

Algorithm	Time Complexity	Main Computational Cost	Space Complexity
ETC	O(1)	Random exploration and fixed exploitation	O(K)
UCB	O(K)	UCB value calculation and max selection	O(K)
Asymptotically Optimal UCB	O(K)	Complex confidence bounds (e.g., KL divergence)	O(K)
TS	O(K)	Posterior sampling and parameter updates	O(K)

4.5 Trade-Off Between Computational Cost and Performance

This study's empirical results highlight a clear trade-off between computational complexity and algorithmic performance:

ETC is the most computationally efficient, with constant-time decisions. However, its simplicity comes at the cost of poor long-term performance, with cumulative regret reaching approximately 25,000 at a horizon of 1,000,000.

Standard UCB offers a solid balance between computational demand and performance. Despite requiring logarithmic and square root operations, its complexity remains manageable. It achieved a cumulative regret of around 12,000, significantly better than ETC.

Asymptotically Optimal UCB introduces more complex confidence interval calculations (e.g., KL-UCB), increasing computation time. Nevertheless, this complexity yields tangible performance improvements—about 12.5% lower regret compared to standard UCB (around 10,500).

Thompson Sampling incurs additional computational overhead from sampling and maintaining posterior distributions. However, this overhead is rewarded with the best overall performance, achieving regret as low as 7,500, a 40% reduction compared to standard UCB.

5 Conclusion

This study's horizon sensitivity analysis reveals critical thresholds that should inform algorithm selection in practical deployments. The logarithmic regret behaviors theoretically predicted for UCB and TS manifest at different horizons—approximately $n \geq 50{,}000$ for TS and $n \geq 100{,}000$ for UCB. This earlier transition for TS represents a significant advantage in applications where user interactions are costly or limited. For short-term deployments ($n < 5{,}000$), the differences between UCB and TS are less pronounced, suggesting that implementation simplicity might reasonably guide algorithm selection in these contexts. Conversely, for long-term deployments ($n > 500{,}000$), the cumulative advantages of TS become substantial, justifying the additional implementation complexity.

The observed performance characteristics have direct implications for recommender system design. In environments with limited initial data, UCB provides reasonable performance with a simpler implementation. For established platforms with high user engagement, TS offers significant long-term regret reduction that could translate to measurable improvements in user satisfaction. While TS provides optimal regret performance, systems with severe computational limitations might benefit from standard UCB, which offers a favorable balance between implementation simplicity and regret minimization. Applications requiring consistent, predictable performance should consider the variance characteristics revealed in Experiment 3, where TS demonstrated not only lower mean regret but also tighter confidence bands. In conclusion, this study's horizon-sensitive analysis demonstrates that algorithm selection for bandit-based recommender systems should be informed not only by theoretical guarantees but also by the expected operational horizon and specific application constraints.

References

1. Zhou, X., Chen, L., Zhang, Y., et al.: 'Deep reinforcement learning for recommender systems: A survey and new perspectives', arXiv:2004.13465, 2020
2. Yu, F., Zhu, Q., Wu, Y., et al.: 'Deep reinforcement learning for recommendation with explicit user-item interactions'. Proc. Int. Joint Conf. Neural Networks (IJCNN), 2020, pp. 1-8
3. Zhou, Y., Li, Y., Tan, V.Y.F.: 'Neural contextual bandits with UCB-based exploration'. Proc. AAAI Conf. Artificial Intelligence, 2020, 34, (04), pp. 6877-6884
4. Wang, H., Wu, Q., Wang, H.: 'Factorization bandits for interactive recommendation: Improvements and extensions', Information Sciences, 2021, 564, pp. 271-286

5. Zhang, W., Liu, Y., Li, Y.: 'Variational Thompson sampling with Gaussian processes'. Advances in Neural Information Processing Systems (NeurIPS), 2021, 34, pp. 21876-21888
6. Lin, C., Wang, Y.: 'Adaptive UCB with dynamic confidence adjustment for non-stationary environments', ACM Trans. Intelligent Systems and Technology, 2022, 13, (4), pp. 1-25
7. Fang, J., Wu, J., Chen, L.: 'Sliding-window Thompson sampling for drifting user preferences'. Proc. ACM Web Conference (WWW), 2023, pp. 3124-3133
8. Liu, X., Zhang, J., Wu, Y., et al.: 'Graph contextual bandits for recommendation'. Proc. ACM Web Conference (WWW), 2021, pp. 1080-1090
9. Wang, S., Zhang, Y., Liu, Q.: 'Transformer-based contextual bandits for sequential recommendation'. Proc. 16th ACM Conf. Recommender Systems (RecSys), 2022, pp. 292-301
10. Chen, H., Li, X., Wang, T.: 'Collaborative Thompson sampling with implicit feedback', ACM Trans. Recommender Systems, 2020, 4, (3), pp. 1-25
11. Huang, Z., Su, C., Li, S., et al.: 'Knowledge-aware contextual bandits for cross-domain recommendation', IEEE Trans. Knowledge and Data Engineering, 2022, 34, (10), pp. 4801-4814
12. Tang, R., Li, M., Zhao, Y.: 'Beyond regret: Multi-objective evaluation for bandit-based recommender systems', Information Processing & Management, 2023, 60, (2), 103198