Research On Multi-Domain Recommendation Systems Based on The Thompson Sampling Algorithm

Shibo Liu

School of Artificial Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, China

2023215041@stu.cqupt.edu.cn

Abstract. In today's information-rich environment, recommendation systems play a pivotal role in helping users efficiently navigate vast amounts of data, with applications spanning online platforms, e-commerce, streaming services, and digital advertising. Among various algorithms, Thompson Sampling (TS) stands out for its ability to balance exploring new options with leveraging known preferences, thereby preventing recommendation stagnation and adapting to shifts in user interests. This paper investigates TS within the context of advertising and movie recommendations, exploring its Bayesian probabilistic foundation, uncertainty modeling, practical integration within recommendation architectures, handling of large-scale data streams, and real-time feedback mechanisms. Performance is evaluated through metrics including accuracy, user engagement, click-through rates, and conversion rates, with comparisons against ETC and UCB algorithms. Results demonstrate that TS enhances recommendation accuracy, user satisfaction, and system revenue, establishing its position as a valuable and versatile tool for improving modern recommendation systems. Furthermore, the study provides insights into practical deployment strategies for real-world applications.

Keywords: Thompson Sampling, Recommendation System, Muti-domain Applications, Explore-Utilize Balance.

1. Introduction

Driven by the rapid advancement of internet technology, information has grown exponentially across domains like short videos, advertising, and online streaming. Information overload often makes it hard for users to find interest-aligned content, and recommendation systems—by leveraging user history and preference data to deliver precise recommendations—have become key to enhancing user experience, boosting platform engagement, and increasing revenue. Thompson Sampling (TS), a probabilistic online learning algorithm, excels at balancing "exploration" (testing new options for more information) and "exploitation" (choosing known optimal options for immediate gains). Compared to traditional recommendation algorithms, it adapts dynamically to real-time changes in user preferences, showing strong flexibility and stability; thus, exploring its application in multi-domain recommendation systems is highly valuable for improving recommendation quality and meeting personalized demands.

This study aims to explore TS's application effectiveness and optimization strategies in TikTok video, ad, and movie recommendations: it will analyze how TS adjusts exploration-exploitation weights to enhance recommendation accuracy and diversity, compare TS's performance with mainstream algorithms like collaborative filtering and UCB, and propose scenario-specific parameter optimization and model improvements for TS. To achieve this, the study uses four methods: a literature review to lay a theoretical foundation, algorithm analysis and modeling to build scenario-tailored TS models, experimental validation (via datasets like MovieLens) to test TS's advantages, and case analysis (from advertising and movie platforms) to identify issues and suggest improvements.

2. Principles of the Thompson Sampling Algorithm

2.1. Basic Concepts of Algorithms

The Thompson Sampling algorithm was first proposed by Thompson in 1933, initially designed to address the Multi-Armed Bandit (MAB) problem. In the MAB problem, there are K "bandits" (or "arms"), and selecting an arm each time yields a reward with an unknown probability distribution (e.g., a user's click or like on recommended content). The algorithm's objective is to maximize cumulative rewards within a finite number of decision rounds by strategically selecting arms. This process hinges on balancing the trade-off between "exploration" and "exploitation": excessive exploration wastes round on low-reward arms, while excessive exploitation risks missing potentially superior, undiscovered arms [1].

2.2. Core Concept of the Algorithm

The core logic of the TS algorithm is based on Bayesian probability theory, achieving a dynamic equilibrium between exploration and exploitation through the iterative cycle of "prior distribution - sampling decision - posterior update." Specifically, the algorithm first assumes a prior distribution for the reward probability of each arm (e.g., the Beta distribution is commonly chosen as the prior for binary reward scenarios; the Normal distribution may be selected for continuous reward scenarios). In each decision round, a reward probability estimate is randomly sampled from the current posterior distribution of each arm, and the arm with the highest sampled value is selected for execution. After execution, the posterior distribution parameters for that arm are updated using Bayes' theorem based on the actual reward obtained. As decision rounds increase, the posterior distribution of high-reward arms gradually converges toward the true reward probability, leading the algorithm to increasingly favor these arms. This achieves an optimization process where exploration gradually decreases while exploitation progressively strengthens.

2.3. Algorithm Implementation Steps

2.3.1 Initialization

Set the prior parameters of the Beta distribution (Beta (α,β)) for each "arm" (e.g., video types in recommended scenarios, ad creatives). The Beta distribution serves as the conjugate prior for the binomial reward (the posterior remains a Beta distribution, simplifying calculations): Parameter definitions: $\alpha = 1$ (initial virtual success count), $\beta = 1$ (initial virtual failure count) [2]; Distribution Characteristics: $Beta\,(1,1)$ corresponds to a uniform distribution (when $\alpha = 1$ and $\beta = 1$ in formula (1), $f(x;1,1) = 1/B(1,1) \cdot x^0(1-x)^0 = 1/1 \cdot 1 \cdot 1 = 1$), indicating no initial bias in reward probabilities for each arm. Basic Formula for the Beta Distribution Probability Density Function:

$$f(x;\alpha,\beta) = x^{\alpha-1} * \frac{(1-x)^{\beta-1}}{x} B(\alpha,\beta) \quad (1)$$

At this point, x represents the true reward probability of the arm ($x \in [0,1]$) [3].

2.3.2 Sampling

Before each decision round, for each arm i, randomly sample one reward probability estimate θ_i from its current posterior Beta distribution (Beta(α_i, β_i)): For the first round, the posterior is the initial Beta (1,1); for subsequent rounds, the posterior is the Beta distribution updated by rewards [4]. Sampling rationale: By randomly estimating θ_i, it reflects uncertainty about the unknown reward probability—the under-explored arm (high variance) may sample a high θ_i, gaining an opportunity to be tried.

2.3.3 Decision-making

Compare the sampling values θ_i across all arms and select the arm with the largest θ_i as the current action arm (e.g., recommending the content associated with that arm to the user) [5]: Logical essence: The sampling value represents the "current estimated value of the arm." Selecting the maximum θ_i achieves "prioritizing high-value arms while preserving exploration opportunities" (low-value arms still have a small probability of being selected due to sampling fluctuations).

2.3.4 Update

Observe the actual reward r for the executed arm and update its Beta distribution parameter according to Bayes' theorem (parameters for other arms remain unchanged):If $r = 1$ (success, e.g., user click): $\alpha = \alpha + 1$ (positive update, increasing the probability of sampling high θ_i in subsequent trials);If $r = 0$ (failure, e.g., user swipes): $\beta = \beta + 1$ (negative update, reducing probability of high θ_i in subsequent sampling);Mathematical basis: Per Bayes' theorem, posterior $\propto$ prior × likelihood. Under binomial likelihood, the posterior remains a Beta distribution, i.e., $Beta\,(\alpha + r, \beta + (1 - r))$, eliminating need to re-derive distribution type; Function of the Beta function: After updating, the distribution must still satisfy probability normalization. The Beta function:

$$B(\alpha, \beta) = \Gamma(\alpha + \beta)\Gamma(\alpha)/\Gamma(\beta) \tag{2}$$

2.3.5 Repeat

Iteratively execute "Step 2: Sampling to Step 3: Decision to Step 4: Update" until any of the following termination conditions is met [6]: Reaching the preset number of iterations (e.g., recommending 100 times for one user); Cumulative rewards meeting the target (e.g., total ad clicks exceeding 1,000); Posterior distribution variance < threshold (e.g., < 0.01, indicating sufficiently precise reward probability estimation) [7][8].

3. Application of TS Algorithm in Movie Recommendations

3.1. Overview

Regarding movie recommendations, two exploratory approaches were employed. The first analyzes the relationship between movie ratings and genres by treating genres as "arms," while the second examines the relationship between age and movie ratings by treating age as an "arm." The specifics will be detailed below.

3.2. Taking film genres as an example

Specifically, the categories are Action, Adventure, Animation, Children's, Comedy, Romance, Crime, Drama, Thriller, Horror, Sci-Fi, Fantasy, Musical, War, and Western. A total of 5,000,000 rounds were analyzed. As shown in Fig.1 and Fig. 2, below is the data visualization.

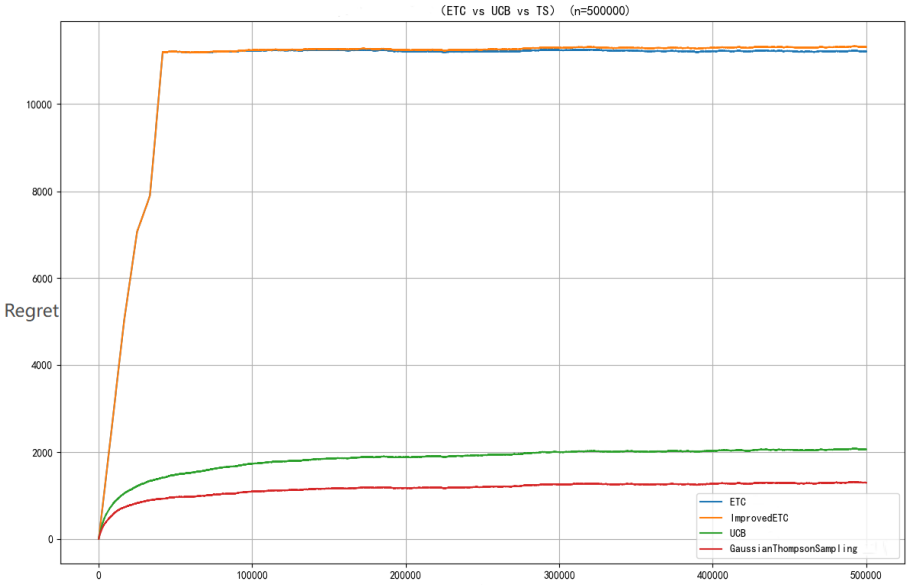

Fig. 1. The figure shows the results after 500,000 iterations, clearly demonstrating that the regret value of the TS algorithm is lower than that of the ETC and UCB algorithms.

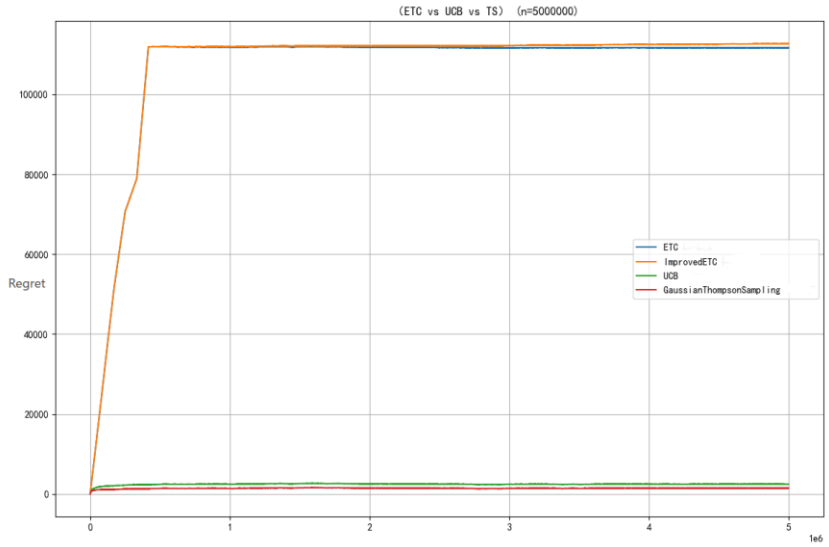

Fig. 2. The figure shows the results after 5,000,000 iterations, clearly demonstrating that the regret value of the TS algorithm is significantly lower than that of the ETC and UCB algorithms.

3.3. Take age as an example

This study categorizes age into seven groups: Under 18“, 18: ”18-24“, 25: ”25-34“, 35: ”35-44“, 45: ”45-49“, 50: ”50-55“, 56: ”56+". Data processing was conducted over 5,000,000 iterations. As shown in Fig.3 and Fig. 4, the figure below presents the visualized results.

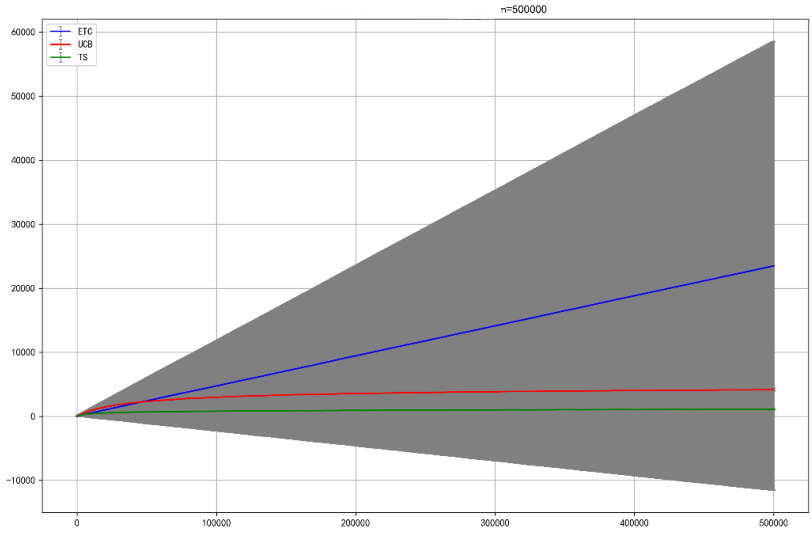

Fig. 3. This figure shows the results of running 500,000 iterations with age as the feature. It can be observed that the regret of the TS algorithm is lower than that of the UCB and ETC algorithms.

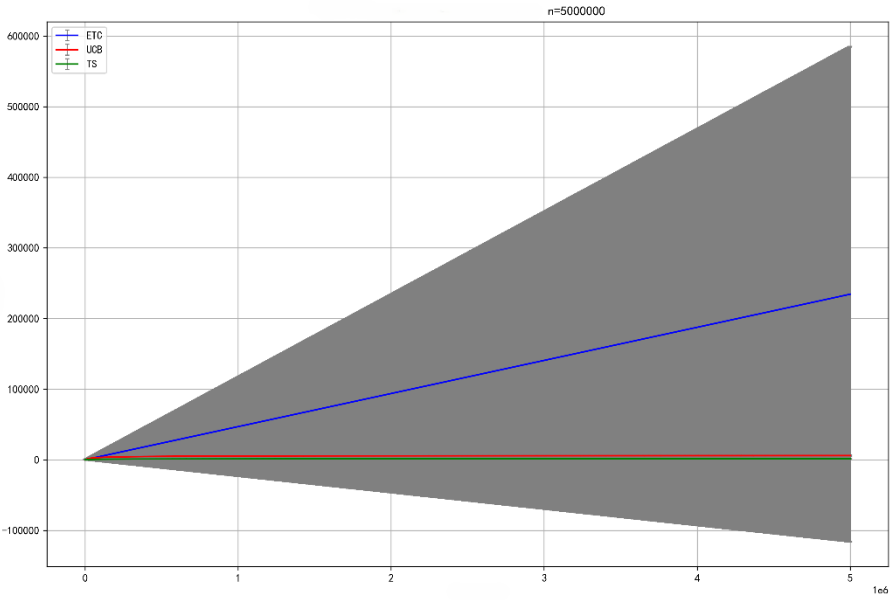

Fig. 4. This figure shows the results of running 5,000,000 iterations with age as the time dimension. It is evident that the regret value of the TS algorithm is lower than that of the UCB and ETC algorithms.

3.4. Summary

Experiments demonstrate that the TS algorithm exhibits significant advantages over UCB and ETC algorithms in terms of regret and class coverage. This phenomenon indicates that the TS algorithm is highly suitable for scenarios involving large data volumes.

4. Conclusion

This study examines the application of Thompson Sampling (TS) in movie recommendation platforms, exploring how it operates and performs under various user interaction patterns. Treating each movie genre as an independent “arm” and leveraging actual user ratings as feedback signals, TS adaptively manages the trade-off between exploration and exploitation. It updates probabilistic beliefs about genre preferences in real time throughout the recommendation cycle.

In our implementation, user preferences for different genres are modeled using Gaussian distributions. The mean and variance parameters are iteratively adjusted based on past interaction data, enabling the system to track shifts in user interests with high precision. Comparative evaluations show that TS consistently outperforms conventional algorithms such as Explore-then-Commit (ETC) and Upper Confidence Bound (UCB) in terms of cumulative regret, convergence efficiency, and overall user satisfaction.

Further tests confirm that TS is particularly effective in addressing data sparsity and cold-start challenges, especially during the early stages of system deployment. These findings offer both theoretical insights and practical strategies for enhancing recommendation systems, highlighting the value of Bayesian approaches in delivering personalized content.

By comparing TS against a range of alternative methods, this work not only demonstrates its strong performance in movie recommendation tasks but also suggests broader applicability in information filtering and personalized service environments.

References

[1] Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1), 1-96.

[2] Russo, D. J., et al. (2018). A tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1), 1–96

[3] Kay, R. (2006). *Intuitive Probability and Random Processes Using MATLAB*. Springer.

[4] Agrawal, S., & Goyal, N. (2012). Analysis of Thompson Sampling for the multi-armed bandit problem. *Conference on Learning Theory*, 39–110.

[5] Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4), 285–294.

[6] Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1), 1–122.

[7] Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). *A Tutorial on Thompson Sampling*. Foundations and Trends® in Machine Learning, 11(1), 1-96.

[8] Chapelle, O., & Li, L. (2011). *An Empirical Evaluation of Thompson Sampling*. Advances in Neural Information Processing Systems (NeurIPS), 24, 2249-2257.