A Comparative Analysis of Multi-Armed Bandit Algorithms in the Research of Optimal Product Discovery in E-Commerce

Tailin Su

{DMT2209225@xmu.edu.my}

Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900, Sepang, Selangor, Malaysia

Abstract: This paper investigates how the Multi-Armed Bandit (MAB) model can maximize the discovery of products of instant noodles based on the Ramen Rating data. The paper will treat recommendation as a sequential decision-making process and will compare three particular algorithms, namely, Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), and Thompson Sampling. Cumulative Regret, Optimal Action Accuracy, and Average Reward were used as measures of performance. Experimental evidence shows that Thompson Sampling is significantly more effective compared to the deterministic strategies. It had the least regret and correctly detected the top-performing brand with a high degree of accuracy in less than 1,000 steps. These results imply that probabilistic methods are a strong and economical method of controlling long-tail inventory in sparse-data settings.

Keywords: Recommender Systems, Multi-Armed Bandit, Thompson Sampling, Cold Start, E-commerce

1. Introduction

These days, recommendation systems have become relevant to e-commerce businesses because they allow them to operate faster in response to the market and achieve higher profits as well. These systems are useful to customers since they do not feel too overwhelmed with the vast library of products available to them. To sellers, effective algorithms help them gain additional users and boost their sales. Nevertheless, conventional techniques such as collaborative filtering have a cold start issue. It is difficult to give recommendations when there are few or even no data on the users. The issue is particularly acute in industries such as the instant food business, where new products come out frequently and can be widely varied in flavor.

This problem has been extensively researched by Multi-Armed Bandit (MAB) algorithms. Silva et al. performed a systematic review and noted that MAB has become a necessity when it comes to the processing of online data streams [1]. They have shown that the conventional batch learning approaches tend to perform poorly in dynamic settings because of high latency whereas MAB algorithms are capable of finding the right balance between exploration and exploitation

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365147

to overcome the issue. Theoretical wise, algorithms keep on being optimized to work in complicated environments. As an example, Liu et al. have recently introduced an extended UCB policy to deal with heavy-tailed reward distributions [2]. Theory has shown that this enhanced approach achieves strong sub-linear regret guarantees, which are much better than common UCB policies under high-variance conditions. Outside the realm of digital content recommendations, MAB has proven extremely flexible in managing physical resources. Decentralized multi-agent bandits were used by Zafar et al. to smartly charge electric vehicles [3]. Empirical outcomes showed that the suggested MAB strategy could decrease charging costs by about 15 per cent over baseline approaches in case of uncertain grids conditions [3]. Nevertheless, the majority of such current studies concentrate on theoretical regret bounds or sophisticated engineering systems such as power grids. Empirical investigations of niche FMCG markets such as instant food, where user preferences are personal, data is limited, and exploration is relatively cheap, are surprisingly scarce.

This gap is filled by the application of the MAB framework to the instant food industry with the help of the Ramen Ratings data set. Rather than making use of complicated hybrid models, individual ramen brands are modeled as the explicit arms of the bandit whereas the user ratings act as the stochastic rewards. Three different algorithms, namely ETC, UCB and Thompson Sampling are adopted and compared to assess their performance. The main objective of this paper is to explore the cumulative regret curves of such algorithms and determine the best approach to optimizing the process of discovering products in a sparse-data e-commerce market.

2. Methodology Overview

2.1 Dataset Source and Preprocessing

Data Source. The data set that has been used in the present research is the Ramen Ratings data set, which was obtained through Kaggle and initially developed by the product review site, The Ramen Rater[4]. The dataset consists of more than 2,500 reviews of instant noodle products shipped out of the Big List. Every record is one product review containing important attributes like brand, product name, style (cup, bowl, or pack), country of origin and stars (quality rating on discrete 0-5 grades). Stars column is the most important attribute in the research study because it reflects the satisfaction level of the user.

Data Preprocessing. The original information may be noisy and sparse, which may prevent the convergence of the algorithms. This study carried out a systematic data preprocessing scheme to create a sound environment for the MAB simulation. First, the rows with non-integer or empty ratings (e.g. the rows with unrated entries) were eliminated to guarantee numerical validity. Moreover, the dataset has a strong long-tail distribution, with most of the smaller brands having either one or two reviews. In order to overcome this sparsity and provide the simulation of a more realistic e-commerce situation with the recommendations of the established brands, the data was filtered to leave only those brands with more than 10 reviews. This is the step that will make sure that every arm has sufficient historical data that the algorithms can use to learn. Lastly, the preprocessed dataset was mathematically associated with the theoretical framework to come up with the MAB formulation. The filtered ramen brands become the available arms in the bandit denoted as K, and the user rating indicated by the stars is the stochastic reward rt. As a

result, whenever an algorithm chooses a brand, it gets a rating that is randomly sampled based on the distribution of the reviews of that particular brand in its history.

2.2 Algorithms and Models

The MAB model was applied in this research to solve the ramen recommendation problem. With the preprocessed data of K brand arms, at any time step t, the algorithm chooses one arm a_t and is rewarded r_t with a reward which is stochastic based on its rating. The iterative nature of this process enables measuring of the cumulative reward over time. Building on this logic, three popular algorithms, ETC, UCB, and Thompson Sampling, were applied to ensure that important performance indicators are compared and finally determine the best strategy to apply to e-commerce applications.

Explore-Then-Commit (ETC). ETC algorithm is the least guided algorithm of all the ones used in bandit problems. The time period T is divided into two stages namely exploration and exploitation. The algorithm behavior is controlled by a pre-established parameter m which indicates the number of exploration rounds given to each arm.

The algorithm does not consider rewards during the exploration stage but instead, only collects data. In the first $m \times K$ time steps (K being the total number of brands and m the number of each arm used), the algorithm picks each of the K arms in a circular manner. This will ensure that after this phase, all the brands have been tested fairly and evenly exactly the same amount of m times which will assist to decide about the best arm and use it during the exploitation phase. On the exploitation phase, when the exploration budget is used up, the algorithm determines the empirical mean reward u_a of each arm a on the basis of the m samples gathered. The arm with the largest estimated average $maxu_a$ is identified, and then in subsequent phases, the algorithm chooses to commit to this analyzed best arm and always use it.

Upper Confidence Bound (UCB). The UCB algorithm is based on the concept of optimism in the face of uncertainty. In contrast to ETC, where exploration and exploitation are divided into distinct stages, UCB balances them together at any point. It builds a confidence interval of the average reward of each arm and constantly chooses the arm with the largest upper limit of that interval.

In every time step t, the algorithm chooses the arm A_t. At time step t, the action selection rule is defined as:

$$A_t = arg_i \max[\, UCB_i(t-1, \delta)] \tag{1}$$

The UCB index is defined as

$$UCB_i(t-1, \delta) = \hat{u}_i(t-1) + \sqrt{\frac{2\log(1/\delta)}{T_i(t-1)}} \tag{2}$$

The formula consists of two components in this context. $\hat{u}_i(t-1)$represents the exploitation stage. This is one of the estimated average rewards of this $arm\ (a)$ given observations until time $t-1$. An increased average motivates the algorithm to take advantage of the best-performing brand at the moment. $\sqrt{\frac{2\log(1/\delta)}{T_i(t-1)}}$ is a measure of uncertainty or an indicator of the width of the confidence interval. It can be referred to as an exploration bonus. The overall sum

of these two terms represents a trade-off inherent in UCB. It is inclined to select arms with a high estimated reward or a large uncertainty, as they are not explored sufficiently.

Mathematicians also remain to validate the theoretical reliability of UCB. As an example, it was recently proven by Ren and Zhang that UCB indices have sharp non-asymptotic regret bounds and thus, this strategy is mathematically sound regardless of the level of exploration [5].

Thompson Sampling. The Thompson Sampling is a Bayesian method, which is quite different than the deterministic approaches such as UCB. Rather than computing a specific upper bound score, Thompson Sampling considers the expected reward of every arm as a random variable [3]. The first step of the data analysis procedure in the ramen context was to normalize the 05 star ratings into a [0,1] range. Because the normalized ratings are not strictly binary but continuous values, the mathematical model of the reward distribution is based on a Gaussian distribution with the unknown mean.To do so, a Normal prior distribution $N(\mu_a, \sigma_a^2)$ is allocated to the expected reward of every brand a, where μ_a estimates the average mean rating and σ_a^2 is the uncertainty of this estimate. At every time step t, the algorithm is not only not picking the brand with the highest empirical average. Rather, it takes one random sample θ_a based on the Gaussian distribution of each brand and picks the one with the maximum sampled value, which is

$$a_t = argmax_a \theta_a \tag{3}$$

After watching the real normalized user rating r_t, the posterior parameters μ_a and σ_a^2 can be updated with iterative updates based on the standard Normal-Normal conjugate Bayesian update rules. The mechanism balances both exploration and exploitation as expected. Should a brand have little information, its variance $\sigma_a^2$is big, which gives a broader distribution that offers a larger probability to generate a large random sample and be explored. With more and more data, the variance decreases, and the algorithm can confidently exploit the best brands. Studies suggest that in sparsely populated data settings, this probabilistic approach tends to be more effective than UCB [1].

2.3 Evaluation Metrics

This research will not use only one metric to accurately gauge the performance of each algorithm. Recent research by Bortko et al. (2025) recommends that the application of a multi-criteria method is necessary to adequately comprehend the trade-offs between exploration and exploitation. To do so, the test mainly focuses on three different evaluation measures, namely, Cumulative Regret, Optimal Action Accuracy, and Average Reward. Cumulative Regret is used as the main measure to compute the opportunity cost of the recommendation process and a lower, flatter regret curve represents the fact that the algorithm has achieved convergence to the optimal choice. Optimal Action Accuracy is the measure of the accuracy of the algorithm and its convergence rate, measured as the proportion of selecting the true best choice of the algorithm, important in developing trust in users in e-commerce apps. Lastly, Average Reward offers an indirect indicator of practical user satisfaction as it is based on average rating levels received, indicating how well the recommendation engine focuses on high-quality products in the specified period.

3. Results

In order to make sure that the evaluation metrics are statistically reliable as well as to address the problem of random initialization in stochastic environments, all the experimental results were obtained through 50 independent Monte Carlo runs. Hence, the reported cumulative regret, optimal action accuracy, and average reward values correspond to the mean averages of these independent runs, which gives a strong and highly objective measure of the actual expected behavior of each algorithm. In this section, data about how the ETC, UCB, and Thompson Sampling algorithms perform on various time horizons will be show. Table 1 shows the details of the experimental results with T=1000.

Table 1. Three algorithms under three evaluation metrics in early exploration phase T=1000.

Algorithm	Cumulative Regret	Optimal Action Accuracy	Average Reward
ETC	261.3297	0.017	0.73466
UCB	240.5454	0.03	0.74927
Thompson Sampling	83.81437	0.656	0.90118

Table 1 illustrates that Thompson Sampling is the best performer at the first step of T=1000 with the minimum cumulative regret of 83.81437, the maximum optimal action accuracy of 0.656, and the maximum average reward of 0.90118. On the other hand, ETC and UCB are observed to have much larger cumulative regret and very low accuracy, which means that they are still deeply entrenched in exploration and cannot see the ideal action.

Then extend the time horizon up to T=2000 to observe the behavior of the algorithms more closely in a long-term perspective, and results are shown in Table 2.

Table 2. Three algorithms under three evaluation metrics in T=2000

Algorithm	Cumulative Regret	Optimal Action Accuracy	Average Reward
ETC	313.4356	0.41	0.836125
UCB	462.3265	0.0395	0.753285
Thompson Sampling	97.77235	0.798	0.937515

As shown in Table 2, the Thompson Sampling continues to perform much better than the rest of the methods, with an extremely low cumulative regret of 97.77235 and a significant increase in its optimal action accuracy to 0.798. It is important to note that the cumulative regret of UCB shoots up to 462.3265, whereas the regret of ETC increases at a slower rate of 313.4356 coupled with a significant improvement in optimal action accuracy to 0.41 indicating that it is moving towards the end of pure exploration.

To sum up, the overall performance indicators at the end of the maximum time horizon, i.e. T=3000, are presented in Table 3.

Table 3. Three algorithms under three evaluation metrics in T=3000

Algorithm	Cumulative Regret	Optimal Action Accuracy	Average Reward
ETC	313.4356	0.606667	0.887817
UCB	675.703	0.049	0.76105
Thompson Sampling	105.2357	0.854	0.95292

As can be seen in Table 3, Thompson Sampling performs the best at T=3000 with a high optimal action accuracy of 0.854 and an average reward of 0.95292. Surprisingly, the total regret of ETC ceases to increase and stays perfectly unchanged at 313.4356, which obviously indicates that it has successfully engaged in an arm and increased its accuracy to 0.606667. On the contrary, UCB has the largest regret of 675.703 and the lowest accuracy improvements, indicating that its exploration strategy is highly inefficient in this particular context.

These evaluation metrics changes over a period of time are shown graphically and examined in Figure 1.

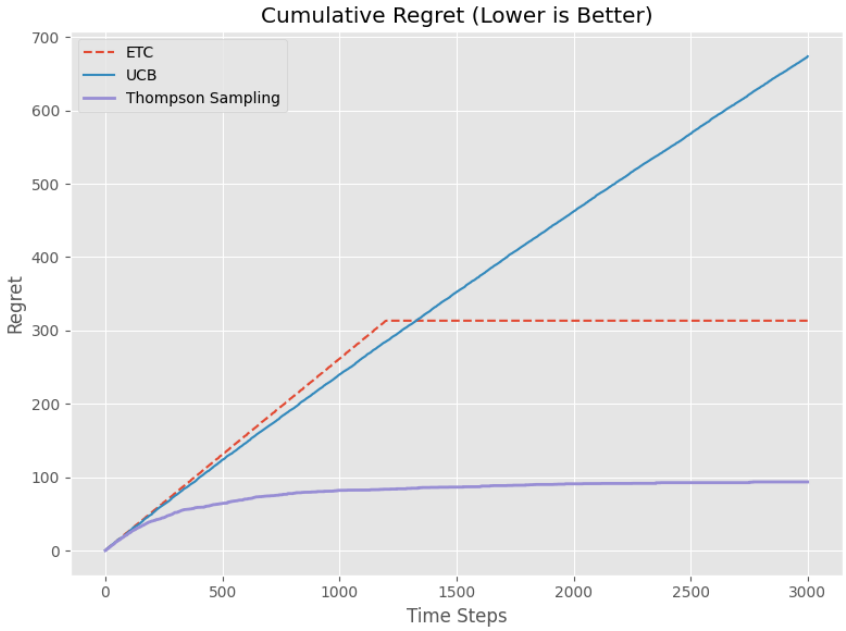

Fig. 1. Cumulative regret trends, optimal action accuracy, and average reward over 3000 time steps of the ETC, UCB and Thompson Sampling algorithms (Photo/Picture credit: Original).

According to Fig. 1, the horizontal axis is the time steps T, which go up to 3000, and the vertical axes are the values of the metric of regret, accuracy, and reward, respectively. Every algorithm is represented by a separate colored line. The Thomson Sampling line stays extremely low and virtually parallel to the bottom axis, which means that it suffers less or no opportunity loss in the cumulative regret subplot. The ETC line experiences a sharp increase initially but entirely levels off as a horizontal line at the end of its exploration stage, and the UCB line exhibits a steady and steep upward trend. Similarly, in the accuracy and reward subplots, the line of the

Thompson Sampling rapidly grows and stabilizes around the maximal possible value, which is another confirmation of the fast convergence and effectiveness of the Thompson Sampling against the much slower and more volatile changes in the ETC and UCB curves that can be clearly seen.

4. Discussion

The experimental findings (Figures 1-3) also indicate a definite performance hierarchy in the problem of ramen recommendations with Thompson Sampling being superior to ETC, and ETC being better than UCB.

In terms of the overall model performance, Thompson Sampling was the most efficient in the present research, as it had the least cumulative regret (≈100.97) and the highest average reward (≈0.95), and its best action accuracy gradually increased to more than 85 percent. Thompson Sampling is based on the use of a Normal prior and posterior update with a Gaussian likelihood to model the reward distribution with an unknown mean, which allows it to naturally balance the trade-off of exploration and exploitation, so that it can converge rapidly and not waste steps on obviously worse products. Such improved flexibility in thin or niche markets is in line with the methodical survey conducted by Silva et al. [1] who established that probability-based methods are highly effective in recommendation settings where fast convergence is necessary. In contrast, ETC had a clear two-stage behavior. Although it eventually became committed to a nearly optimal brand after about 1,200 steps (achieving 60% accuracy), the main drawback was the excessively high sunk cost when it first had to explore brands randomly. Although the deterministic exploration ensures that convergence will happen eventually, their inflexible nature tends to give suboptimal short term user experience.

Although UCB and Thompson Sampling can reach similar asymptotic performance in theory, UCB showed rather poor results in this particular simulation, with the best accuracy of the optimal actions being only 0.049. It is not an indication of an implementation failure, but it indicates an acute weakness in the classical UCB algorithm when used in scenarios with a large number of actions (K=57) and a somewhat small time horizon (T= 3000). Namely, applying the default exploration parameter without domain-specific adjustment makes the confidence intervals become too loose. Therefore, the deterministic UCB algorithm is mathematically constrained to indefinitely sample each one of the underexplored brands to minimize these large uncertainty boundaries. In the extremely restricted 3000 time steps, the algorithm remains stuck in an endless exploration cycle, never reaching any real exploitation stage. This pattern of behaviour proves that classical UCB has theoretical restrictions in high dimensional, short horizon problems, which was theoretically shown by Liu et al. [2] and subsequently placed into context with non-asymptotic bounds introduced by Ren and Zhang [5].

As a way of improving the performance of the best performing model, which is Thompson Sampling, considering the addition of contextual features as an optimization approach can be adopted. The inclusion of contextual information (demographics of users, their regional taste, or their purchase history) into the recommendation system has proven to be particularly beneficial in terms of minimizing first-time uncertainty and offering highly customized forecasts, as it was shown by Chen [6] in related fields of recommendations. To optimize UCB and ETC, the adoption of de-centralized or tuned multi-agents strategies, like in the model used

by Zafar et al., in complex smart systems [3], might dynamically adapt the scope of exploration depending on their variation and thus efficiently remove the penalty of searching clearly worse arms.

In spite of the obvious algorithmic knowledge, the shortcomings of this paper are mainly rooted in the fact that the simulated environment is fixed and does not have real-time user feedback. The simulation also takes it as a given that the underlying reward probabilities, which are assumed to be representative of user preferences of ramen brands, do not change with time. Nevertheless, it has recently been demonstrated through the development of recommendation systems that offline assessments do not necessarily reflect the non-stationary nature of the preferences of actual users because the tastes of users and market trends are changing dynamically and evolve over time [7]. This dynamic preference evolution, even with non-linear forgetting of past interests, can make existing static models ineffective within a live commercial setting where ongoing adaptation is necessary [7].

Considering these constraints, the next step in future studies is to generalize this framework to more realistic, real-world situations. Increasing the data set to incorporate the multi-dimensional user attributes and implementing the algorithms into live A/B test settings will enhance the generalization abilities of the models to large extents. Moreover, investigating non-stationary bandit models, especially adaptive forms of Thompson Sampling that have been explicitly created to respond to changing reward probabilities, would be an extremely useful addition [8]. Also, the use of Large Language Models (LLM) to provide information about Multi-Armed Bandit strategies is a state-of-the-art solution to dealing with such non-stationary conditions since the recommendation engine can adaptively perceive the semantic changes in user preferences that are complex in nature [8]. Such a strategy is similar to the urgent necessity to track the dynamics in changing conditions, which was highlighted by Bortko et al. [10] Finally, closing the gap between imaginary bandit modeling and real-life e-commerce recommendation engines.

5. Conclusion

The research was intended to discover the most effective algorithmic strategy for product recommendation in niche e-commerce markets, where data sparsity often challenges conventional collaborative filtering approaches. By implementing the recommendation process as a Multi-Armed Bandit problem on the Ramen Ratings dataset, the trade-offs between exploration and exploitation across three common algorithms were effectively measured.

The findings are very encouraging regarding the effectiveness of Thompson Sampling. In contrast to the ETC algorithm, which is prone to a fixed separation between learning stages, and the UCB algorithm, which might be too conservative with regard to exploration, Thompson Sampling uses Bayesian probability to respond dynamically to user feedback. Experiments proved that Thompson Sampling had the smallest cumulative regret and the largest average reward so it was the best option to maximize user satisfaction at real time.

Nevertheless, this paper has weaknesses. It was assumed that the user preference is stationary, but in practice, they may vary with time. The Non-stationary Bandit algorithms can be used to represent drifting user interests in future research. Also, using Contextual Bandits based on user

metadata (such as geographic location, spice tolerance) might add more personalization to the recommendation and enhance its accuracy.

To sum it all up, an e-commerce platform that focuses on niche goods and has a small amount of historical information can implement a Thompson Sampling-based recommendation engine as a strong, mathematically based solution to the cold start problem with the ultimate aim of increasing sales and retaining customers.

References

[1] Silva, N., Werneck, H., Silva, T., Pereira, A. C. M., & Rocha, L.: Multi-Armed Bandits in Recommendation Systems: A survey of the state-of-the-art and future directions. Expert Systems With Applications. 197, pp. 1-15 (2022)

[2] Liu, K., Zheng, T., & Zhou, Z. H.: Extended UCB Policies for Multi-armed Bandit Problems. Machine Learning. 115(1), pp. 7-7 (2025)

[3] Zafar, S., Feraud, R., Blavette, A., Camilleri, G., & Ahmed, H. B.: Decentralized multi-agent multi-armed bandits for smart electric vehicles charging. Engineering Applications of Artificial Intelligence. 163(P6), pp. 113088 (2026)

[4] Kaggle: Dataset resources: Ramen Ratings. Kaggle Datasets. pp. 1-1 (2026)

[5] Ren, H., & Zhang, C. H.: On Lai's upper confidence bound in multi-armed bandits. Annals of Mathematical Sciences and Applications. 10(2), pp. 381-404 (2025)

[6] Chen, Y.: Contextual bandits to increase user prediction accuracy in movie recommendation system. ITM Web of Conferences. 73, pp. 1-10 (2025)

[7] Ding, H., Zhu, W., Hu, G., & Bu, Z.: Capturing Dynamic User Preferences: A Recommendation System Model with Non-Linear Forgetting and Evolving Topics. Systems. 13(11), pp. 1034 (2025)

[8] Qi, H., Guo, F., & Zhu, L.: Thompson Sampling for Non-Stationary Bandit Problems. Entropy. 27(1), pp. 51-51 (2025)

[9] de Curto, J., de Zarza, I., Roig, G., Cano, J. C., Manzoni, P., & Calafate, C. T.: LLM-Informed Multi-Armed Bandit Strategies for Non-Stationary Environments. Electronics. 12(13), pp. 2814 (2023)

[10] Bortko, K., Gudowicz, M., Śniegowski, S., & Świder, A.: How to Tame a Multi-Armed Bandit? A Multi-Criteria Decision Analysis Approach to Bayesian Adaptive Exploration in Non-Stationary Environments. Procedia Computer Science. 270, pp. 5350-5359 (2025)