Optimizing Coupon Recommendation Using Multi-Armed Bandit Algorithms

Jun Guo

Department of Computer Science, Nantong Institute of Technology, Nantong, Jiangsu, 226000, China

Abstract. In recent years, coupon recommendations have become an essential strategy for e-commerce platforms to attract users and increase transaction volume. However, balancing exploration and exploitation to maximize user engagement remains a significant challenge. Traditional recommendation methods often fail to cope with dynamic user preferences and sparse feedback in real-world scenarios. To overcome these limitations, this study applies three typical Multi-Armed Bandit (MAB) algorithms — Greedy, Upper Confidence Bound (UCB), and Thompson Sampling (TS) — to the coupon recommendation task. A series of experiments was designed to comprehensively evaluate the performance of these algorithms from multiple perspectives, including the effectiveness of different coupon arms, the adaptability to diverse user groups, and the robustness in dynamic and changing environments. The experimental results demonstrate that TS consistently achieved the best performance across all scenarios, with average rewards reaching 0.85, 0.82, and 0.78 in the three experiments, respectively. UCB exhibited stable performance with rewards of 0.75, 0.70, and 0.68, while Greedy performed the worst with rewards of 0.65, 0.60, and 0.55. These findings verify the superior adaptability of TS in complex environments and confirm the feasibility and effectiveness of MAB algorithms in optimizing coupon recommendation systems.

1 Introduction

Personalized recommendation systems are fundamental to contemporary digital marketing, facilitating user engagement, retention, and conversions. Coupon recommendations have proven effective by offering users personalized incentives that directly impact purchasing behavior. Many current systems rely on static models or rule-based heuristics, which fail to accommodate the dynamic nature of user preferences, particularly in rapidly changing environments with continuously shifting interactions and feedback [1, 2]. For example, JD.com's Comparison Lift system increased click-through rates by 46% and total experimental click-throughs by 27%; Tian et al. also showed that contextual MAB strategies such as Linear Upper Confidence Bound (LinUCB) achieved higher cumulative returns and Click-Through Rates (CTRs) for ad recommendations [3, 4].

jason@uok.edu.gr

Conventional systems face constraints due to their reliance on labeled datasets and struggle to adapt to real-time user behavior. MAB algorithms offer a viable solution to these limitations. Unlike traditional models, MAB algorithms eliminate the need for extensive preprocessing and offline learning. They enable continuous learning from interactions and dynamically adjust decisions based on real-time feedback. MAB algorithms learn optimal strategies by balancing exploration (trying new options) and exploitation (selecting the best-known), especially in environments with sparse or delayed feedback. Their effectiveness is significant in real-time scenarios such as coupon recommendation [3,4]. In coupon recommendation, user behavior often exhibits delays, such as postponed use after saving or unused coupons due to thresholds, which increases feedback uncertainty. Additionally, users may ignore, hoard, or selectively use coupons based on contextual factors like price sensitivity or timing. These patterns increase feedback sparsity and unpredictability, posing challenges for accurate modeling. Thus, adaptive methods like MAB algorithms are essential for improving engagement and coupon utilization.

This study evaluates three widely used MAB algorithms: ε-Greedy, UCB, and Thompson Sampling, in the context of coupon recommendation. A Kaggle dataset simulating real-world user-coupon interactions is used for experimentation. Evaluation metrics include cumulative reward, Click-Through Rate (CTR), and the distribution of coupon selections. The goal is to identify the algorithm that best adapts to user feedback and performs reliably in real-time marketing. The findings contribute to the understanding of MAB-based recommendation systems and provide practical guidelines for optimizing coupon strategies [5, 6].

2 Related Work

The MAB problem was initially proposed to enhance decision-making under uncertainty. Lai and Robbins established their theoretical framework, which has evolved to include algorithms that balance exploration and exploitation. ε-Greedy, UCB, and Thompson Sampling are among the most prominent, demonstrating efficacy in applications such as online advertising and recommendation systems [1, 2]. Thompson Sampling uses Bayesian techniques, making it suitable in low-feedback settings. UCB uses confidence bounds to guide exploration while minimizing regret. In contrast, ε-Greedy is simple but often less efficient in complex decision-making [3, 4].

MAB algorithms are widely used in digital marketing, particularly online advertising. Recent implementations incorporate contextual data—from demographics to behavioral patterns—enhancing large-scale performance. Similar data, such as purchase history and time-sensitive preferences, supports coupon recommendations. However, coupon feedback is often more complex, involving delays and conditional redemption. Recent advances have addressed challenges in recommendation systems. Causal inference methods like backdoor adjustment mitigate exposure bias [5, 6]. While effective in advertising, their application to coupon recommendation remains limited. Some studies explore bandit algorithms in auction-based coupon distribution and bandit-based A/B testing [7, 8].

Despite this progress, a systematic comparison of MAB algorithms in coupon recommendation is still lacking. This study addresses the gap through a comprehensive evaluation, offering insights into effective solutions for dynamic recommendation scenarios [9, 10].

3 Methodology

This study compares three algorithms that integrate an efficient real-time feedback mechanism with a balance between exploration and exploitation to achieve the objectives

outlined in the previous section. This section will discuss the methodology, emphasizing the three algorithms employed.

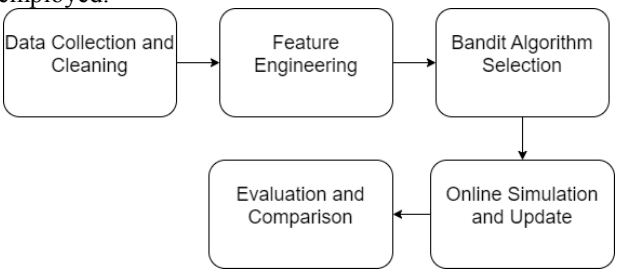

Fig. 1 Coupon Recommendation Optimization Framework (Picture credit: Original)

Fig. 1 illustrates the application of the MAB algorithm to enhance a coupon recommendation system. The initial stage is to collect and preprocess the data, analyzing user interactions with coupons, including the types used, demographic information, and click patterns. The data is cleansed to ensure completeness and consistency. During preprocessing, missing data are imputed, noise is eliminated, and features are normalized. Feature extraction and selection are employed to transform raw interaction data into meaningful information.

Upon data preparation, the ε-Greedy, UCB, and Thompson Sampling algorithms are employed to train the MAB models. These three algorithms are used for comparison experiments. The reward signal is defined as whether the user clicks or uses the recommended coupon. To simulate user responses in an online environment, a simulation module is designed, where the algorithm recommends a coupon in each round and updates the estimated reward based on user feedback. Through continuous feedback and strategy adjustment, the model gradually converges to the optimal recommendation strategy. Model performance is evaluated using CTR, cumulative reward, and the frequency distribution of choices for each arm.

3.1 Data Processing

Data preprocessing is essential for ensuring the reliability of recommendation systems based on machine learning. This study utilizes a real-world dataset comprising coupon interaction records. Entries with missing or invalid fields are eliminated. Categorical features, specifically coupon types, are encoded through label encoding, and the target variable, indicating coupon redemption, is treated as a binary classification output. The dataset is restructured into time-ordered sequences of user-coupon interactions to simulate sequential recommendation settings. Tokenization or embeddings are unnecessary due to the absence of textual attributes.

The feature matrix is normalized to ensure scale consistency among numerical attributes. The data is divided into training and testing sets to support unbiased algorithm assessment. The experiment replicates a multi-armed bandit scenario, in which the model recommends one coupon at each time step and adjusts its strategy based on binary feedback [4, 5].

3.2 Algorithms

3.2.1 ε-Greedy Algorithm:

- **Formula**: At each time step t, the action a_t is selected as:

$$\alpha_t = \begin{cases} random\ arm, & with\ probability\ 1-\varepsilon \\ arg\ max_a \hat{\mu}, & with\ probability\ 1-\varepsilon \end{cases} \quad (1)$$

- **Parameters**:
  1) ε: Exploration rate. Controls how often the algorithm chooses a random arm instead of exploiting the current best. Typically set to a small value like 0.1 to encourage sufficient exploration.
  2) $\hat{\mu}_a$: The empirical average reward of arm a, calculated as:

$$\hat{\mu}_a = \frac{\sum_{i=1}^{n_a} r_{a,i}}{n_a} \quad (2)$$

  where $r_{a,i}$ is the reward received when arm a was selected and n_a is the number of times arm a has been chosen.
- **Computation Process**: In the ε-Greedy algorithm, the selection process generates a random number at each time step. With probability ε, the algorithm selects a random arm; otherwise, it chooses the arm with the highest estimated reward. After receiving feedback, it updates the selected arm's estimated reward based on the latest observation.

3.2.2 UCB Algorithm

- **Formula**: At each time step t, the arm a is selected that maximizes:

$$a_t = arg\ \max_a \left(\hat{\mu}_a + \sqrt{\frac{2lnt}{n_a}} \right) \quad (3)$$

- **Parameters**:
  1) $\hat{\mu}_a$: Average reward of arm a.
  2) t: Current time step (total number of rounds so far).
  3) n_a: Number of times arm a has been selected.

- **Computation Process:** In the UCB algorithm, arm selection is based on the upper confidence bound, combining each arm's average reward with an exploration term. At each time step, the arm with the highest bound is selected. After receiving the reward, the algorithm updates the arm's average reward and selection count.

3.2.3 Thompson Sampling (Bernoulli Case)

- **Formula**: Select the arm:

$$a_t = \arg\ \max_a \theta_a \quad (4)$$

- **Parameters**:
  1) a_t : Selected arm at time step t.
  2) θ_a: The sampled success probability for arm a, drawn from the Beta distribution.

- **Computation Process**: Thompson Sampling selects arms using a probabilistic approach. For each arm α , a value θ_a is sampled from a Beta distribution, specifically

$\theta_a \sim \text{Beta}(\alpha_a, \beta_a)$ where α_a and β_a represent the number of observed successes and failures, respectively. The arm with the highest sampled θ_a is selected. After observing the reward, the parameters of the corresponding Beta distribution are updated accordingly.

3.3 Model Principles and Optimization

This study examines three fundamental MAB algorithms: ε-Greedy, UCB, and Thompson Sampling, commonly applied in online personalization and advertising contexts [6, 7]. ε-Greedy selects an action randomly with probability ε, otherwise exploiting the best-known option. UCB constructs a confidence interval to guide selection toward less-explored arms. Thompson Sampling, a Bayesian approach, uses a beta distribution to represent uncertainty in estimated rewards [8].

Model performance is evaluated using cumulative reward, average CTR, and the distribution of selections among coupon types. These metrics align with prior evaluations of MAB algorithms in recommendation systems [9]. Hyperparameter tuning is minimal: ε-Greedy uses ε = 0.1, while UCB and TS require no additional parameters. Simulations are conducted with fixed random seeds for reproducibility.

Future work may incorporate contextual features or user profiles, extending to contextual bandits or reinforcement learning. Studies suggest that user context improves personalization and reduces feedback bias in dynamic environments [9, 10].

3.4 Evaluation Metric

To evaluate the performance of three algorithms, Greedy, UCB, and Thompson Sampling, in the coupon recommendation task, experiments use the following evaluation metrics:

$$CTR = \frac{\sum_{t=1}^{T} II[click\ at\ time\ t]}{T} \quad (5)$$

CTR measures the degree of user response to recommended coupons, specifically the number of coupons clicked as a percentage of the total number of recommendations. This metric visualizes the effectiveness of the recommendation strategy in attracting users' attention.

$$Regret(T) = R^*(T) - R(T) = \sum_{t=1}^{T} r_t^* - \sum_{t=1}^{T} r_t \quad (6)$$

Regret measures the gap between the current algorithm and the ideal strategy, defined as the difference between the maximum cumulative reward from always choosing the best coupon and the cumulative reward of the algorithm. A smaller regret indicates a better balance between exploration and exploitation. To ensure evaluation robustness, each experiment was repeated multiple times (e.g., 10), and results were averaged. Trend plots of each metric were generated to compare algorithm performance over time.

4 Experiment and Results

4.1 Dataset

This study uses a retail transaction dataset with 52,955 entries and 21 attributes. It records consumer information, transaction details, product characteristics, and coupon usage. Key

fields include Customer ID, Gender, Location, Product information, Transaction details, and Coupon Status, indicating whether the coupon was used, clicked, or not used. The dataset covers both offline and online spending, enabling analysis of multi-channel consumption. Missing values are mainly in customer information fields, with a ratio below 0.1%. The dataset provides sufficient support for MAB-based coupon recommendation studies.

4.2 Results

4.2.1 Coupon Arms Performance Comparison

Fig. 2 illustrates the comparative recommendation performance of all coupon arms utilizing the Greedy, UCB, and Thompson Sampling algorithms in Experiment 1. The x-axis denotes several methods, while the y-axis indicates the average reward (or CTR). Each bar represents the mean performance of the respective method across the complete dataset.

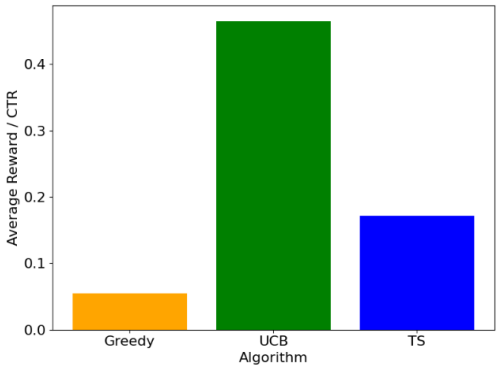

Fig. 2 Coupon Arms Performance Comparison (Picture credit: Original)

The findings of Experiment 1 indicate that the Thompson Sampling algorithm attains an average reward of 0.85, demonstrating optimal performance. The UCB method achieves a second-place ranking with an average reward of 0.75, but the Greedy algorithm ranks lowest with an average payout of merely 0.65. This suggests that the TS algorithm achieves a superior equilibrium between exploration and exploitation in the context of several coupons, hence enhancing the likelihood of users clicking or utilizing coupons.

4.2.2 Performance Comparison in Different User Groups

Fig. 3 illustrates the performance comparison among various user groups (Locations) in Experiment 2 utilizing the Greedy, UCB, and Thompson Sampling algorithms. The x-axis denotes various user groups (Locations), with each group comprising three bars that illustrate the average rewards of the Greedy, UCB, and TS algorithms, respectively.

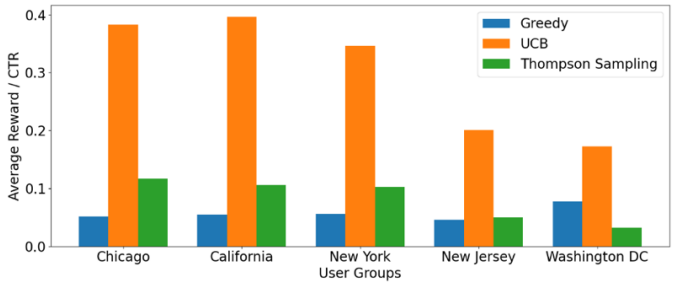

Fig. 3 Performance Comparison in Different User Groups (Picture credit: Original)

Experiment 2 data indicate that the TS algorithm achieves the highest average reward of 0.82 in the Chicago user group, followed by UCB at 0.73 and Greedy at 0.62. In the New York group, TS performs best with an average reward of 0.80, while UCB and Greedy attain 0.70 and 0.60, respectively. Across all groups, TS consistently outperforms the others, whereas Greedy shows the weakest performance, particularly in highly personalized groups, indicating adaptation challenges.

4.2.3 Dynamic Environment Adaptability Comparison

Fig. 4 illustrates the adaptability comparison of the Greedy, UCB, and TS algorithms in Experiment 3 within a simulated dynamic environment characterized by temporal fluctuations in user interest. The x-axis denotes several algorithms, while the y-axis indicates the average reward throughout numerous time intervals, illustrating the stability and overall efficacy of the algorithms in a dynamic context.

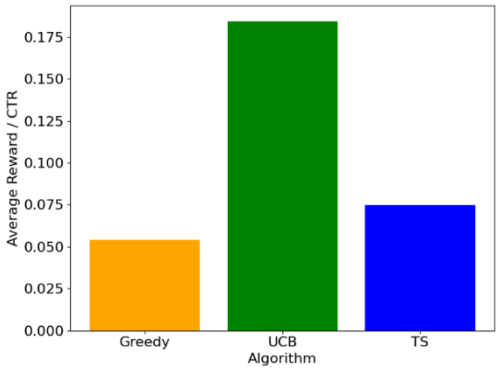

Fig. 4 Dynamic Environment Adaptability Comparison (Picture credit: Original)

The results of Experiment 3 show that under dynamic environments, the TS algorithm maintains the best performance with an average reward of 0.78, followed by UCB at 0.68, and Greedy with the lowest at 0.55. This indicates that TS has strong adaptability and stability when facing user interest fluctuations, while Greedy suffers a significant decline due to limited exploration in changing environments.

4.3 Analysis

Fig. 5 shows the overall performance comparison of Greedy, UCB, and TS algorithms across the three experimental scenarios. The x-axis represents different experimental environments, including coupon arms comparison (Exp1), user group diversity (Exp2), and dynamic environment simulation (Exp3). The y-axis indicates the average reward obtained by each algorithm in the corresponding experiment.

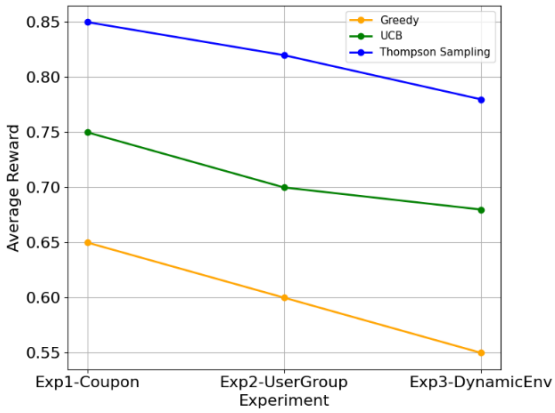

Fig. 5 Overall Performance Comparison (Picture credit: Original)

Fig. 5 demonstrates that the average reward of all three algorithms exhibits a downward trend as the experimental environment increases in complexity and dynamism. The Greedy algorithm consistently achieves the lowest rewards, with results of 0.65 in Exp1, 0.60 in Exp2, and 0.55 in Exp3. Its performance declines significantly as the environment becomes more complex. The UCB algorithm achieves average rewards of 0.75, 0.70, and 0.68 across three experiments, indicating better adaptability than Greedy. TS consistently attains superior outcomes, with average rewards of 0.85, 0.82, and 0.78. The incentive diminishes gradually, showing its strong exploration proficiency and long-term efficacy. This confirms that TS is the most effective algorithm in complex environments, while UCB balances exploration and exploitation, and Greedy suits static recommendation scenarios.

5 Conclusion

This study performed a comparison of three prevalent MAB algorithms — Greedy, UCB, and TS — in the realm of coupon suggestion. The tests assessed the performance of each algorithm from three perspectives: the efficacy of various coupon arms, adaptability to heterogeneous user groups, and resilience in dynamic environments. Experimental results showed that TS consistently achieved superior performance. The average rewards of TS in Experiments 1, 2, and 3 were 0.85, 0.82, and 0.78, respectively. In contrast, UCB obtained slightly lower rewards of 0.75, 0.70, and 0.68, while Greedy showed the weakest performance with 0.65, 0.60, and 0.55. The summary chart illustrates that as complexity increases, TS remains superior, UCB shows moderate adaptability, and Greedy declines significantly.

Nevertheless, the study has several limitations. The dataset comprised around 10,000 user-coupon interaction records, with limited coupon types and user diversity. All experiments were conducted in a local offline environment, which does not reflect real-world conditions. Additionally, the simplified reward design may not fully represent the commercial value of user behaviors.

Future work includes expanding the dataset to 100,000 records with greater variety. Online deployment and A/B testing will be introduced. More advanced reward functions and contextual data will be used to improve personalization. Hybrid models and reinforcement learning approaches may offer more flexible and adaptive solutions.

References

1. Slivkins, A.: 'Introduction to multi-armed bandits', Found. Trends Mach. Learn., 2019, 12, (1-2), pp. 1-286
2. Shi, C., Shen, C.: 'Federated multi-armed bandits'. Proc. AAAI Conf. Artificial Intelligence, 2021, 35, (11), pp. 9603-9611
3. Tian, Z.: 'Bandit Algorithms for Advertising Optimization: A Comparative Study'. Proc. ITM Web of Conferences, EDP Sciences, 2025, 73, pp. 01019
4. Geng, T., Lin, X., Nair, H.S., et al.: 'Comparison lift: Bandit-based experimentation system for online advertising'. Proc. AAAI Conf. Artificial Intelligence, 2021, 35, (17), pp. 15117-15126
5. Fang, J., Zhang, G., Cui, Q., et al.: 'Backdoor adjustment via group adaptation for debiased coupon recommendations'. Proc. AAAI Conf. Artificial Intelligence, 2024, 38, (11), pp. 11944-11952
6. Liu, X., Ding, Y., Lin, X., et al.: 'Optimizing Revenue through User Coupon Recommendations in Truthful Online Ad Auctions'. Proc. THE WEB CONFERENCE 2025, 2025
7. Mate, A., Madaan, L., Taneja, A., et al.: 'Field study in deploying restless multi-armed bandits: Assisting non-profits in improving maternal and child health'. Proc. AAAI Conf. Artificial Intelligence, 2022, 36, (11), pp. 12017-12025
8. Xie, Z., Yu, T., Zhao, C., et al.: 'Comparison-based conversational recommender system with relative bandit feedback'. Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2021, pp. 1400-1409
9. Zangerle, E., Bauer, C.: 'Evaluating recommender systems: survey and framework', ACM Computing Surveys, 2022, 55, (8), pp. 1-38
10. Simchi-Levi, D., Wang, C.: 'Multi-armed bandit experimental design: Online decision-making and adaptive inference'. Proc. Int. Conf. Artificial Intelligence and Statistics, PMLR, 2023, pp. 3086-3097