A Strategy for Advertisement Placement based on the Multi-Armed Tiger Problem

Yibo Zhao

The Pennsylvania State University, 201 Old Main, University Park, PA 16802, USA

yxz5956@psu.edu

Abstract. The purpose of this study is to investigate the effectiveness of using Multi armed bandit model, which contains ϵ -greedy, Upper Confidence Bound (UCB), and Thompson sampling algorithms, to optimize online advertisement placement. Through simulating different types of ad placements using different algorithms and comparing them, this paper intends to demonstrate the feasibility of the Multi-Armed Robber model for the ad placement problem. The results show that the multi-armed bandit model can improve the ad click rate compared with the traditional ad placement strategy. Thompson sampling algorithm outperforms the ϵ -greedy algorithm and UCB algorithm in this paper's experiments, which can better balance exploration and application and reduce regret. The algorithm provides a more efficient method of allocating ad resources. These findings provide new insights into the field of digital marketing and may have an impact on the development of actual ad placement strategies.

Keyword: Ad Placement, Multi armed bandit, Thompson Sampling, ϵ -greedy, Upper Confidence Bound.

1. Introduction

Traditional media (also known as “old media”) such as television, radio, newspapers and magazines have been used for marketing purposes for decades. While these mediums are still a common way to reach customers and other companies, new media are significantly changing the way people access information as the number of Internet users worldwide reaches 5.35 billion by 2024, with the average user spending 6.5 hours a day online [1]. The rapid rise of new media forms such as online advertising, online news and social media has greatly expanded the breadth and depth of information dissemination. However, users are faced with an extremely large amount of information every day, which often leads to information overload, making it difficult for advertisements to effectively reach the target audience [1].

To address this challenge, traditional online ad delivery methods, such as rule-based systems and simple A/B testing, have gradually shown their limitations. For example, Auer, Cesa-Bianchi, and Fischer pointed out in their study that these methods often lead to suboptimal resource allocation and delayed assessment of ad effectiveness due to their inability to dynamically adapt to changes in user behavior and preferences [2]. To solve these problems, researchers proposed Multi-Armed Bandit (MAB) algorithm as an effective optimization tool.

Geng, Lin and Nair experimentally investigated the application of the MAB algorithm in target advertising audience evaluation, and the results showed that the algorithm significantly improved the click-through rate and conversion rate of the advertisements [3]. In addition, Hu et al. explored the performance of the MAB algorithm in dealing with the mean-variance setting, emphasizing the advantages of this algorithm in dealing with uncertainty and optimal resource allocation[4].Auer, Cesa-Bianchi and Fischer further investigated the application of the MAB algorithm in display ad optimization, noting that this algorithm can significantly improve ads by updating user data in real time to significantly improve ad effectiveness [5].

Cesa-Bianchi, Gentile and Zappella showed that the application of MAB algorithm in real-time bidding and ad targeting can effectively balance the relationship between exploring new strategies and utilizing known strategies in a dynamic environment, thus optimizing the allocation of advertising resources [6, 7].Jiang, on the other hand, explored the MAB model in personalized online advertising, revealing the advantages of the algorithm in handling high-dimensional user data and coping with dynamic user engagement, which is important for the accuracy and effect optimization of advertisement placement [8].

However, despite these studies demonstrating the long-term benefits of the MAB algorithm, the existing literature still lacks a systematic comparison of its performance at the initial stage of ad delivery. For example, Zhou pointed out that although the ε-greedy algorithm is widely used due to its simplicity, a fixed exploration rate may lead to poor performance in the long run when faced with rapidly changing environments [9]. Zhou et al. showed that the UCB algorithm is able to better adapt to changes in advertisement effectiveness without the need for a preset exploration rate, but its computational complexity is high, especially sensitive to the initial parameters [10]. In contrast, the Thompson sampling algorithm can more accurately handle uncertainty and adapt to dynamic environments by sampling from posterior probability distributions, but Jiang and Shabalina pointed out that this method is computationally burdensome when accurate estimation of the posterior distribution is required [11].

To address this research gap, this study aims to compare the performance of ϵ-greedy, UCB, and Thompson sampling methods in the initial stages of advertisement placement through simulation. Through these comparisons, advertisers are able to reduce the decision-making risk and optimize the click-through rate and conversion rate of advertisements in the early stages of ad placement, thus improving the ROI of advertisements [12].

The aim of this study is to explore the application of dobby slot machine algorithms in online ad placement, especially the comparison of ϵ-greedy, UCB and Thompson sampling methods. By simulating an ad placement scenario, we will analyze in detail the performance of these algorithms at the initial stage of ad placement. The research methodology includes ad placement simulations using real datasets and a series of experiments to compare the effectiveness of the three algorithms. Our goal is to reveal which algorithm can better balance exploration and utilization in different ad placement scenarios, so as to provide effective decision support for advertisers [10-12].

2. Methodology

This section will define the ad placement problem and discuss the issues that would arise in that scenario. It will also introduce the three algorithms of multi-armed bandit, discuss their advantages and disadvantages, and how to solve the problems encountered in the advertisement placement scenario.

2.1. Definition of ad placement

Advertising placement is a business strategy that aims to increase awareness and sales of a product or service by attracting potential customers through targeted advertising content. In the digital advertising space, the problem is particularly complex because advertisers need to maximize their return on investment by showing the most appropriate ads to the right audience at the right time within a limited budget. There are three main challenges to this problem, which are as follows

1. Variety of advertising choices: Advertisers usually have multiple advertising choices, each with different potential returns and costs. The central question in this challenge is which ads to choose to maximize returns and minimize costs.

2. Uncertainty in user behavior: There is a high degree of uncertainty as to whether a user will click on an advertisement or interact with its content. The central question is how to increase the probability of users clicking on the ads, i.e., CTR, and the probability of users interacting with the ads, i.e., conversion rate.

3. Dynamic market conditions: The multidimensionality, diversity and rapidity of the market environment and user preferences require advertisements to be able to shift and adjust quickly according to changes in the market and users.

Therefore, the problem of how to effectively manage advertising resources, i.e., selecting the best combination of advertisements from a large number of possible advertisements to maximize the achievement of a specific objective (e.g., click-through rate, conversion rate, or ad revenue), is faced by online advertisement placement. In previous studies, there are many approaches to optimize online advertisement placement, among which especially multi-arm bandit models and machine learning. In the study of Jiang, C., & Shabalina, O. it was shown how these algorithms can optimize the ad placement strategy, especially to improve the effectiveness and ROI of the ads in the face of uncertainty [10]. In a study by Kong, S. T. the application of dobby slot machine algorithms combined with machine learning in advertising campaigns is discussed [11]. In a study by Jiang, C. it was shown that MAB algorithms can effectively improve the click-through rate and conversion rate of advertisements, especially in the rapidly changing online advertising market [12]. But on the other hand, the risk and cost of advertisers in the initial stage of placing ads is great. Therefore, the goal of this paper is to optimize the click-through rate at the initial stage of online advertisement placement.

2.2. Multi armed bandit

The multi-armed bandit (MAB) problem is a decision model for maximizing reward or optimizing outcomes in the presence of multiple options. In the MAB problem, the decision maker chooses between multiple options (or “arms”), each with an unknown probability distribution representing the probability of obtaining a reward.

The central challenge of the MAB model is to balance the two strategies of “exploration” and “exploitation”:

- Exploration phase: Attempts to obtain information about each option by selecting different options, so as to know which options are likely to deliver the best rewards.
- Exploitation phase: Based on the information gained, the arm with the known higher return is selected to maximize the total return.

For online ad placement problems, the Multi-Arm Bandit (MAB) model provides an efficient solution because it directly optimizes the core challenges in ad placement. For example, dynamic decision support, balanced exploration and utilization, budget efficiency maximization, simplification of complex decision-making processes, and real-time feedback utilization can better adapt to the dynamic changes in the advertising market and improve the precise targeting of advertisements in the ad placement problem [6]. And several studies have shown the effectiveness of MAB algorithms in improving the click rate and conversion rate of advertisements as well as the potential to optimize resource allocation in dynamic advertising environments, and MAB algorithms are able to be effective in dynamic bidding environments that can effectively bid on advertisements and allocate resources [1,2,3,5]. In addition to this, Contextual Bandit algorithm provides a new approach that can significantly improve the relevance and user engagement of advertisements [4]. And for complex markets and multidimensional users, the MAB model excels in handling high-dimensional user data and dynamic user engagement [7].

2.3. ϵ -Greedy

ϵ -Greedy is a straightforward multi-armed bandit strategy that centers on a trade-off between exploration and exploitation. The strategy is controlled by the parameter ϵ. The main feature of the ϵ -Greedy strategy is that it is simple to implement and easy to understand, but its difficulty lies in choosing an appropriate value of ϵ to balance the efficiency of exploration and exploitation. Too high a value of

ϵ may lead to too much ineffective exploration, while too low a value of ϵ may lead to prematurely ignoring potentially better choices. In addition, since the exploration rate is fixed, the algorithm performs poorly in dealing with rapidly changing market environments, which may lead to unsatisfactory results in the long run. In dynamic environments, this fixed strategy may lead to sub-optimal resource allocation [8].

Pseudocode:

```
for t = 1, 2, 3, ... do
    # Exploration vs. Exploitation decision
    Generate r ~ U(0, 1)
    if r > ε then
        # Exploitation step: select the arm with the highest estimated reward
        A_t ← argmax(Q_k) for k = 1, ..., K
    else
        # Exploration step: select a random arm
        A_t ← random choice from {1, 2, ..., K}
    end if
    # Pull the selected arm and observe the reward
    r_t ← reward from pulling arm A_t
    # Update the estimated reward for the selected arm
    N[A_t] ← N[A_t] + 1
    Q[A_t] ← Q[A_t] + (1 / N[A_t]) * (r_t - Q[A_t])
end for
```

r is from the uniform distribution □(0,1) in which a random number is generated to decide whether to explore or not.

ε is the exploration rate, if the random number r is greater than ε, the current optimal arm is selected for utilization, otherwise an arm is randomly selected for exploration.

A_t is the arm selected at time step t.

r_t is the reward received from arm A_t.

Q[A_t] is the expected reward estimate for the updated arm A_t.

2.4. Upper Confidence Bound (UCB)

The upper confidence interval (UCB) strategy is a method of making decisions based on the upper confidence interval. For each selection, UCB considers the average gain for each arm and the uncertainty of selecting that arm (usually based on the number of times that arm has already been selected), and then selects the arm with the highest upper confidence interval limit.The main advantage of UCB is that it balances exploration and exploitation in a systematic way, automatically adjusting the intensity of exploration by taking into account the uncertainty of each arm. This makes UCB very effective in applications where the overall number of trials needs to be minimized in order to quickly converge to the optimal selection, but it is more computationally complex than Epsilon-Greedy and may be too aggressive for arms with high volatility returns. In a study by Jin Zhou (2024) it is shown that UCB is better able to adapt to changes in advertising effectiveness without the need for a preset exploration rate, and it performs especially well in the face of volatile markets. However, it is more sensitive to initial parameters [8].

In UCB we select the arm by this formula:

$$A_t = \arg\max_a \left(\overline{X}_a + \sqrt{\frac{2\log t}{N_a}} \right) \tag{1}$$

A_t: This is the action or "arm" chosen at time t. It represents the decision made by the algorithm about which arm to pull based on the calculated upper confidence bounds.

$\overline{\mathrm{X}}_{\mathrm{a}}$: This is the average reward obtained from arm a up to time t. It measures the effectiveness of the arm based on past performances.

N_{a}: This is the number of times arm a has been selected up to time t. This count is used to ensure that all arms are sufficiently explored.

$\sqrt{\frac{2\log t}{\mathrm{N}_{\mathrm{a}}}}$: This term represents the uncertainty or the confidence interval for arm a. It ensures that arms that have not been explored as much as others are given a chance to prove their potential. The uncertainty decreases as N_{a} increases, meaning the more an arm is tested, the more precise its estimated value becomes

Pseudocode:

```
Initialize Q_k = 0, N_k = 0 for all k = 1, ..., K
for t = 1, 2, 3, ... do
    # Selection step
    for k = 1, ..., K do
        if N_k == 0 then
            A_t ← k
        else
            UCB_k ← Q_k + sqrt(2 * log(t) / N_k)
        end if
    end for
    A_t ← argmax(UCB_k) for k = 1, ..., K
    # Pull the selected arm and observe the reward
    r_t ← reward from pulling arm A_t
    # Update the number of times arm A_t has been pulled
    N[A_t] ← N[A_t] + 1
    # Update the estimated reward for arm A_t
    Q[A_t] ← Q[A_t] + (1 / N[A_t]) * (r_t - Q[A_t])
end for
```

Q_k is the current estimated expected reward for arm k.

N_k is the number of times arm k has been selected.

UCB_k is the Upper Confidence Bound value of arm k.

At each time step t, the algorithm selects the arm with the largest UCB_k and pulls that arm to observe the reward r_t.

The number of times an arm is selected N_k and the expected reward Q_k are then updated.

2.5. Thompson Sampling

Thompson Sampling (TS), also known as posterior sampling or probabilistic matching, is an efficient Bayesian multi-armed bandit (MAB) strategy. The strategy solves the exploration-exploitation tradeoff problem by combining Bayesian statistical inference and probabilistic sampling. Specifically, the Thompson sampling method builds a probabilistic model for the payoffs of each "arm" that is typically updated based on historical data. Each time an action is chosen, the algorithm draws a sample from the current posterior distribution for each arm. Then, the arm with the highest sample payoff is selected for placement. The key aspect of this approach is that each sample may yield different results, which naturally introduces exploration, especially for arms with less data (and therefore higher uncertainty). This strategy has proven to be very effective, especially in situations where the dynamics of the environment change or where there is a high demand for both initial exploration and long-term utilization.A paper was shown that this algorithm performs well in handling uncertainty and adapting to dynamic advertising markets, but that its computational complexity is relatively high [9].

In the Thompson sampling algorithm, we first need to initialize our prior distribution:

$$\mathrm{Beta}(\alpha_a = 1, \beta_a = 1) \tag{2}$$

Then sample from the Beta distribution:

$$\theta_a(t) \sim \mathrm{Beta}(\alpha_a, \beta_a) \tag{3}$$

Then select the arm:

$$A_t = \arg\max_a \theta_a(t) \tag{4}$$

Then observe and update the distribution:

$$\alpha_{A_t} = \alpha_{A_t} + r_t \tag{5}$$

$$\beta_{A_t} = \beta_{A_t} + (1 - r_t) \tag{6}$$

These steps encapsulate a Bayesian updating mechanism that constantly updates the prior distribution of each robotic arm based on observed rewards, and repeating these steps in the algorithm adjusts its estimate of the probability of success for each arm.

Pseudocode:

```
Initialize α_k = 1, β_k = 1 for all k = 1, ..., K
for t = 1, 2, 3, ... do
    # Sample from the Beta distribution for each arm
    for k = 1, ..., K do
        θ_k ~ Beta(α_k, β_k)
    end for
    # Selection step
    A_t ← argmax(θ_k) for k = 1, ..., K
    # Pull the selected arm and observe the reward
    r_t ← reward from pulling arm A_t
    # Update the parameters of the Beta distribution for arm A_t
    if r_t == 1 then
        α[A_t] ← α[A_t] + 1
    else
        β[A_t] ← β[A_t] + 1
    end if
end for
```

α_k and β_k are the parameters of the Beta distribution corresponding to each arm k.

θ_k is the sample value of each arm drawn from the Beta distribution.

A_t is the arm selected at time step t. r_t is the sample value obtained from arm A_t.

r_t is the reward from arm A_t, usually binary (0 or 1).

Update the Beta distribution parameters of the selected arm A_t with the value of the reward r_t: if the reward is 1, update α_k; if the reward is 0, update β_k.

3. Experiment

The purpose of this experiment is to demonstrate the effectiveness of the multi-armed bandit model by simulating advertisement placement and to compare the performance of ϵ -greedy, UCB and TS algorithms.

The data source for the experiment is Dataset: Online Advertisement Click-Through Rates published on mendeley data. there are 11 features in the data. they are Age, gender, income, location, ad type, ad topic There are 11 features in the data: Age, gender, income, location, ad type, ad topic, ad placement, clicks, click time, conversion rate, and Click Through Rate.

Experimental Steps:

1. experimental design:

- Determine the Arm of the experiment such as ad type, ad topic, incom level.

- The arm chosen for this experiment is ad type, and each ad type is an Arm.

2. algorithm implementation:

- Implement ϵ -greedy, UCB and Thompson sampling algorithms using Python.

- Ensure that the algorithms can adjust the ad placement strategy in real time based on the sampled data.

3. simulation:

- Sample the dataset to simulate the ad placement within a set number of rounds. The next Arm is selected by the algorithm by giving the algorithm's recommended arm based on the sampled data.

5. Result Evaluation:

- Algorithm performance is reflected by calculating the cumulative regret value for each algorithm

Regret = Best Arm Bonus - Selected Arm Bonus

Because of the large uncertainty at the beginning of the MAB model, each algorithm was run ten times to take the average and express the standard deviation.

4. Result

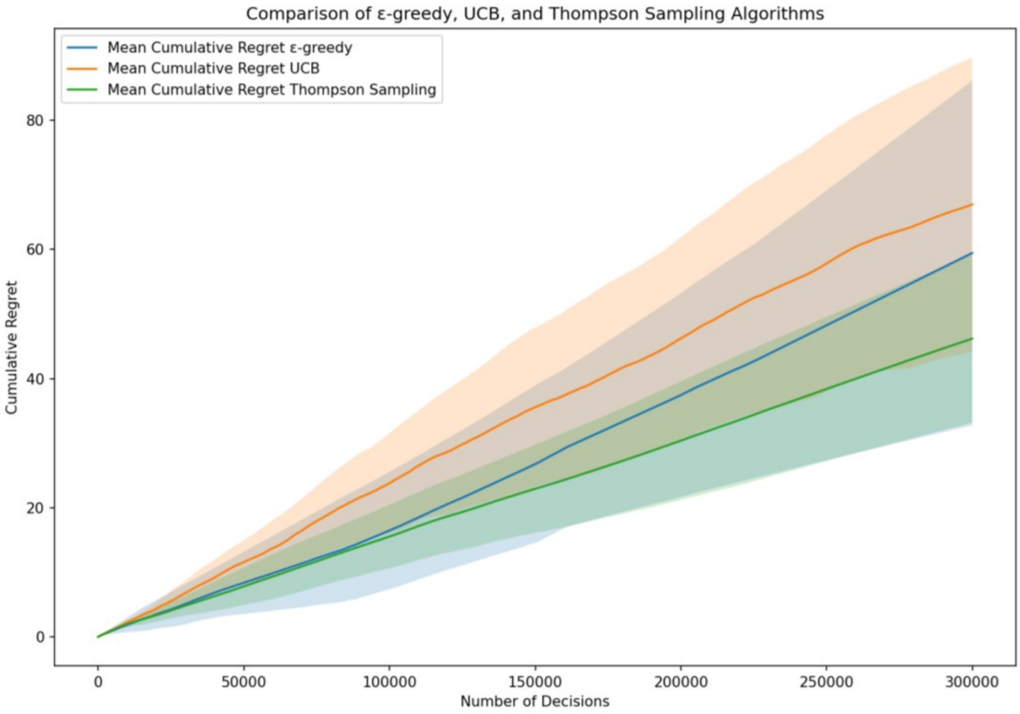

Figure 1. Number of rounds Vs cumulative regrets

Graphic Analysis

Algorithm comparison:

UCB (orange line) shows the highest cumulative regret in the graph, indicating that it performs the worst in this particular setup. ε-greedy (blue line) cumulative regret is in the middle of the spectrum and outperforms UCB. Thompson Sampling (green line) demonstrates the lowest cumulative regret, suggesting that it works the best in these strategies.

Confidence intervals:

The shaded areas in the figure represent the confidence intervals for each algorithm, indicating the variability of the algorithm's results. Thompson Sampling shows the least variability, indicating that its results are the most stable. UCB has the widest confidence intervals, indicating that its performance fluctuates widely between trials.

Performance Discussion:

The high regret of the UCB algorithm may indicate that its exploration mechanism is not efficient enough in the face of such problems, or that the parameter settings (e.g., exploration coefficients) are not well adapted to the characteristics of the data. This may have led to over-exploration of suboptimal arms or failure to utilize the best known arms in a timely manner. ε-greedy algorithms, although simple, have their performance strongly affected by the exploration probability ε. The performance of the UCB algorithm may be significantly affected by the probability of exploration ε. Appropriate values of ε may significantly improve its performance, but it usually lacks the flexibility to adapt to dynamic environments. Thompson Sampling selects actions by sampling from a posteriori distributions, and this probability-based approach seems to be more effective in dealing with uncertainty and the trade-offs between exploration and exploitation, especially when the rewards have a high degree of variability.

Thus, Thompson Sampling shows its power on the Multi-armed bandit problem, especially in terms of consistently optimizing performance and reducing regrets over long runs. ucb may require parameter tuning or modifications to the algorithm itself to better accommodate specific experimental setups or reward distributions. epsilon-greedy's performance sits somewhere in between the other two algorithms in this experiment, the However, it still has a large standard deviation, so it is not stable. ϵ -greedy and UCB's performance both depend on the setting of the parameters. This is difficult to apply to new placement scenarios that do not have too much data.

5. Conclusion

In summary, this study shows that Thompson Sampling is the most suitable algorithm among three Multi armed bandit algorithms for the optimization of advertisement placement strategies. It not only reduces cumulative regret, but also maintains a stable performance. The UCB algorithm Although it possesses good theoretical properties for exploring and exploiting the balance, finer tuning parameters are needed in practical applications to realize its potential. The ε-greedy algorithm Although simple, its performance is limited, especially in dynamic environments that require long running times.

However, the MAB model still has some limitations in the problem of advertisement placement. How to balance exploration and utilization in practice is a limitation, if too much exploration will increase regret, while not enough exploration will miss the best Arm. In addition, the fast changing market and the responsiveness of the MAB model is another test. In practice, the responsiveness of the Multi-Arm Bandit model may not be fast enough to adjust strategies in real time. In addition, the core of the MAB model is to optimize timeliness metrics such as click-through rate. This may lose sight of advertisers' long-term business goals. In addition, there are many limitations in the experiments of this paper, such as whether the data source of the database is trustworthy, and there will be errors if the data volume is too small.

Future work could consider investigating more variants of UCB, as well as further optimizing the parameter settings of Thompson Sampling, with a view to achieving better results in more complex and dynamic advertising environments. Additionally by investigating more types of arms and different advertising scenarios, as well as combining different datasets. Studying the performance of these algorithms in different problems can provide more basis for actual advertisement placement.

References

[1] Geng,T,Lin,X.,&Nair,H.S.(2019).Online evaluation of audiences for targeted advertising via bandit experiments.

[2] Auer, P.,Cesa-Bianchi, N., & Fischer, P. (2021). A Novel Application of Multi-Armed Bandits to Optimize Display Advertising.

[3] Hu,H.,Charpentier,A.,Ghossoub,M.,& Schied,A.(2022).The multi-armed bandit problem under the mean-variance setting.

[4] Auer,P,Cesa-Bianchi,N., & Fischer, P. (2021). Enhancing Online Advertisements Through Contextual Bandit Algorithms.

[5] Cesa-Bianchi, N., Gentile, C., & Zappella, G. (2020). Adaptive Strategies in Real-Time Bidding Based on Multi-Armed Bandit.

[6] Cesa-Bianchi, N., Gentile, C., & Zappella, G. (2022). Algorithmic Enhancements in Multi-Armed Bandits for Sophisticated Ad Targeting.

[7] Jiang, C. (2021). Multi-armed bandits for personalized online advertising.

[8] Zhou, J. (2024). Application and comparative analysis of adaptive strategies in multi-armed bandit algorithms. Applied and Computational Engineering, 64(1), 237-248.

[9] Zhou, Z., Srikant, R., Veeravalli, V. V., Milenkovic, O., & Mehta, P. (2021). Impact of Thompson sampling in programmatic ad buying.University of Illinois Urbana-Champaign.

[10] Jiang, C.,& Shabalina, O. (2022). Using advanced bandit algorithms to optimize ad placement on social media. University of Illinois Urbana-Champaign.

[11] Kong, S. T. (2020).Multi-armed bandits and machine learning for effective ad campaigns. University of Illinois Urbana-Champaign.

[12] Jiang,C.(2019).Real-time optimization of online advertisements using MAB models. University of Illinois Urbana-Champaign.