Dotanet: Ad Simulation of Multi-Platform Firm

Fatemeh Abbasian-Abyaneh

August 18, 2024

Abstract

This report presents an in-depth analysis of advertisement simulation for a multi-platform firm using Multi-Armed Bandit (MAB) algorithms, specifically Upper Confidence Bound (UCB), Thompson Sampling (TS), and a novel Adaptive Exploration-Exploitation (AEE) strategy. We explore the performance of these algorithms in estimating click probabilities and maximizing cumulative revenue under varying conditions, characterized by parameters like the number of firms (K), sites (L), and dimensions (d). The report also compares the performance of UCB, Thompson Sampling, and AEE, discussing the implications of each approach, how model parameters influence the results, and potential areas for improvement.

Contents

1 Introduction

In the digital advertising industry, firms often need to decide how to allocate their ad budgets across multiple platforms (sites) to maximize engagement (clicks) and revenue. This allocation problem can be modeled as a Multi-Armed Bandit (MAB) problem, where each platform is akin to an "arm" of a slot machine. The challenge lies in balancing exploration (trying out different platforms) with exploitation (continuing to invest in platforms that yield high returns).

This report investigates two popular MAB algorithms, UCB and Thompson Sampling, along with a newly proposed Adaptive Exploration-Exploitation (AEE) strategy, in the context of ad allocation. We simulate the performance of these algorithms under different conditions, analyze the results, discuss the impact of varying key parameters, and propose potential areas for further improvement.

2 Methodology

2.1 Problem Formulation

Consider a scenario with K firms, each aiming to advertise across L different platforms (or sites). The effectiveness of an ad for firm i on site j is quantified by a click probability p_{ij}, which depends on the interaction between the firm's features and the site's features.

Let:

- K: Number of firms.
- L: Number of sites.
- d: Dimensionality of the feature space.
- c_i: Feature vector for firm i.
- s_j: Feature vector for site j.

The click probability p_{ij} is given by:

$$p_{ij} = c_i \cdot s_j$$

where c_i and s_j are normalized to lie on a d-dimensional sphere.

2.2 UCB Algorithm

The Upper Confidence Bound (UCB) algorithm selects the platform that maximizes the upper confidence bound of the estimated click probability. The idea is to prioritize platforms with higher uncertainty (less explored) to balance exploration and exploitation.

The UCB value for platform j is computed as:

$$\text{UCB}_j = \hat{p}_j + \alpha \sqrt{\frac{\log t}{n_j}}$$

where:

- $\hat{p}_j$: Estimated click probability.
- n_j: Number of times platform j has been selected.
- α: Exploration parameter.
- t: Current time step.

UCB has a strong theoretical foundation, providing guarantees on its performance in terms of regret minimization. However, in practice, its performance can be sensitive to the choice of the exploration parameter α. If α is too large, UCB might over-explore, leading to slower convergence; if too small, it may under-explore, missing out on potentially lucrative platforms.

2.3 Thompson Sampling

Thompson Sampling (TS) selects the platform based on sampling from the posterior distribution of the click probabilities. It balances exploration and exploitation by considering the uncertainty in the probability estimates.

At each step, a click probability is sampled from the Beta distribution for each platform:

$$\theta_j \sim \text{Beta}(a_j, b_j)$$

where a_j and b_j are updated based on the observed clicks and non-clicks for platform j.

Thompson Sampling is known for its practical effectiveness, often outperforming UCB in real-world scenarios. Its ability to adaptively balance exploration and exploitation makes it robust across various settings. However, the computational complexity of updating and sampling from the posterior distribution increases with the number of platforms and firms, which can be a limitation in large-scale problems.

2.4 Adaptive Exploration-Exploitation (AEE) Strategy

The Adaptive Exploration-Exploitation (AEE) strategy is a hybrid approach designed to dynamically adjust the balance between exploration and exploitation based on real-time performance. Unlike UCB, which has a fixed exploration term, and Thompson Sampling, which relies on probabilistic sampling, AEE uses performance feedback to modulate exploration intensity.

2.4.1 Concept of AEE

AEE operates by monitoring the cumulative reward (or revenue) and adjusting its exploration-exploitation trade-off accordingly. If the cumulative reward growth slows down, AEE increases exploration to discover potentially better platforms. Conversely, if the reward growth is steady or increasing, AEE reduces exploration to focus on exploitation.

This dynamic adjustment makes AEE potentially more flexible and responsive to changing environments than either UCB or Thompson Sampling. The key challenge in AEE lies in setting the appropriate thresholds for when and how to adjust exploration. Incorrect thresholds could either lead to excessive exploration, wasting resources, or insufficient exploration, missing out on better opportunities.

2.5 Implementation Details

The implementation of the UCB (Upper Confidence Bound), Thompson Sampling, and Adaptive Exploration-Exploitation (AEE) strategies was executed using Python. This section outlines the structure and logic used to simulate these algorithms and analyze their performance in a multi-platform advertising environment.

2.5.1 Setup and Initialization

The simulation environment was carefully designed to model the interaction between multiple firms (K) and various advertising platforms or sites (L). The key components of the setup include:

- **Click Probabilities (c)**: For each firm, a click probability vector c_i is initialized, representing the likelihood that an ad from firm i will receive a click when placed on a given site. These probabilities are generated randomly to simulate diverse levels of ad effectiveness across different firms and sites.
- **Site Traffic (s)**: Site traffic, denoted by s_j, represents the relative user engagement on each platform. Traffic values are generated using an exponential distribution to reflect varying degrees of platform popularity and are subsequently normalized. This ensures that the model accurately reflects the differences in user interaction across the sites.
- **Feature Vectors (d)**: The dimensionality d of the feature space represents the characteristics of both firms and sites. Feature vectors for firms and sites are randomly generated and normalized to lie on a d-dimensional sphere. This normalization allows for consistent feature interactions across the simulation, which are crucial for determining click probabilities.

2.5.2 Algorithmic Design

The algorithms implemented in this study—UCB, Thompson Sampling, and AEE—each follow a unique approach to balance exploration and exploitation. Below is an overview of how each algorithm is designed to operate within the simulation framework:

Upper Confidence Bound (UCB) The UCB algorithm operates by selecting the site that maximizes the upper confidence bound of the estimated click probability. This approach is particularly useful in environments where the algorithm needs to explore less-frequented sites while still exploiting those that are known to be profitable. The confidence bound is calculated using both the estimated click probability and an exploration term that diminishes as the number of selections for a particular site increases. Over time, as more data is collected, the algorithm adjusts its focus, gradually shifting from exploration to exploitation.

Thompson Sampling (TS) Thompson Sampling is implemented to balance exploration and exploitation by sampling from the posterior distribution of the click probabilities for each site. At each iteration, the algorithm samples a click probability for each site from a Beta distribution, which is updated based on observed clicks and non-clicks. This probabilistic approach allows Thompson Sampling to adaptively explore different sites, particularly those with higher uncertainty, while also exploiting sites that show high potential for generating clicks.

Adaptive Exploration-Exploitation (AEE) The AEE strategy is designed as a hybrid approach that dynamically adjusts the balance between exploration and exploitation based on real-time performance. This strategy combines elements of UCB and Thompson Sampling, allowing it to adapt its behavior according to the observed outcomes. By continuously evaluating the performance of different sites, AEE can shift focus between exploration and exploitation in a more nuanced way, potentially outperforming traditional methods in complex environments.

2.5.3 Simulation Process

The simulation process involves running each algorithm over a specified horizon, where firms repeatedly decide which sites to allocate their ads to, based on the strategies outlined above. Throughout the simulation, data is collected on key metrics, including estimated click probabilities, cumulative revenue, and revenue per ad. These metrics are then used to evaluate and compare the performance of the algorithms under various conditions, such as different values of K, L, and d.

The results of these simulations are visualized using Python's data visualization libraries, allowing for a clear comparison of how each algorithm performs across different scenarios. This approach provides insights into the strengths and weaknesses of each method and helps identify the conditions under which one algorithm may be more effective than the others.

3 Results

In this section, we present the results of our simulations using the UCB, Thompson Sampling, and AEE algorithms under different conditions. We analyze their performance in terms of estimated click probabilities, cumulative revenue, and revenue per ad.

3.1 Performance of UCB

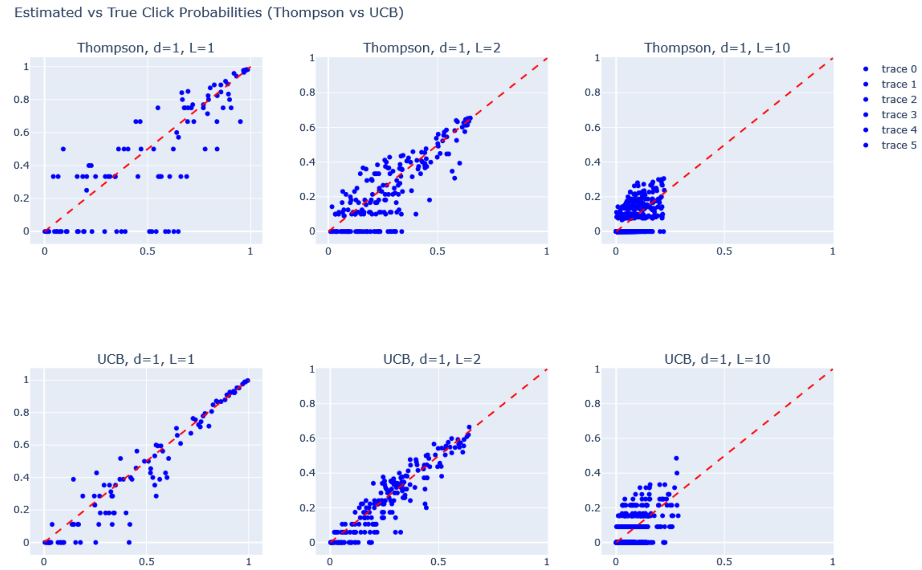

Figure 1: Estimated vs True Click Probabilities (UCB)

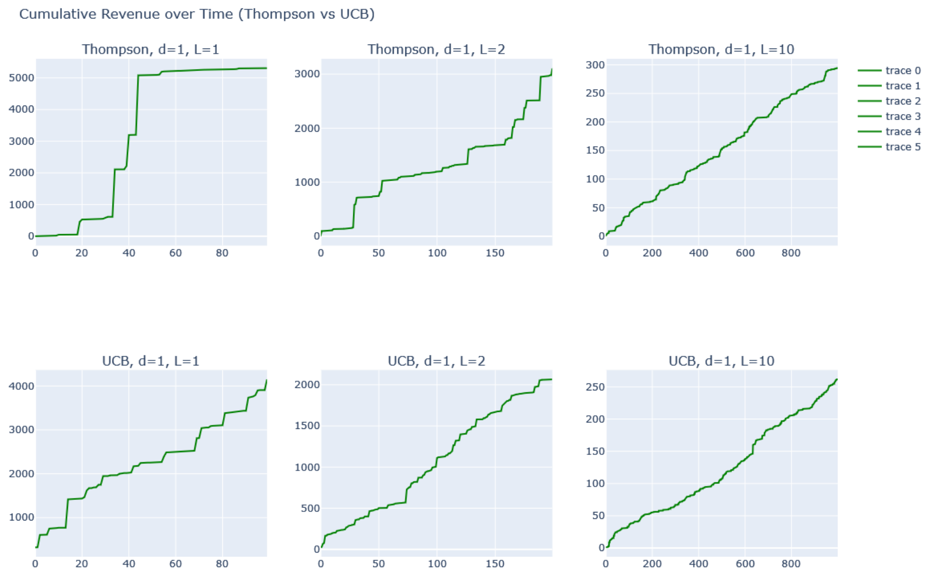

Figure 2: Cumulative Revenue over Time (UCB)

Figure 3: Revenue per Ad (UCB)

3.2 Performance of Thompson Sampling

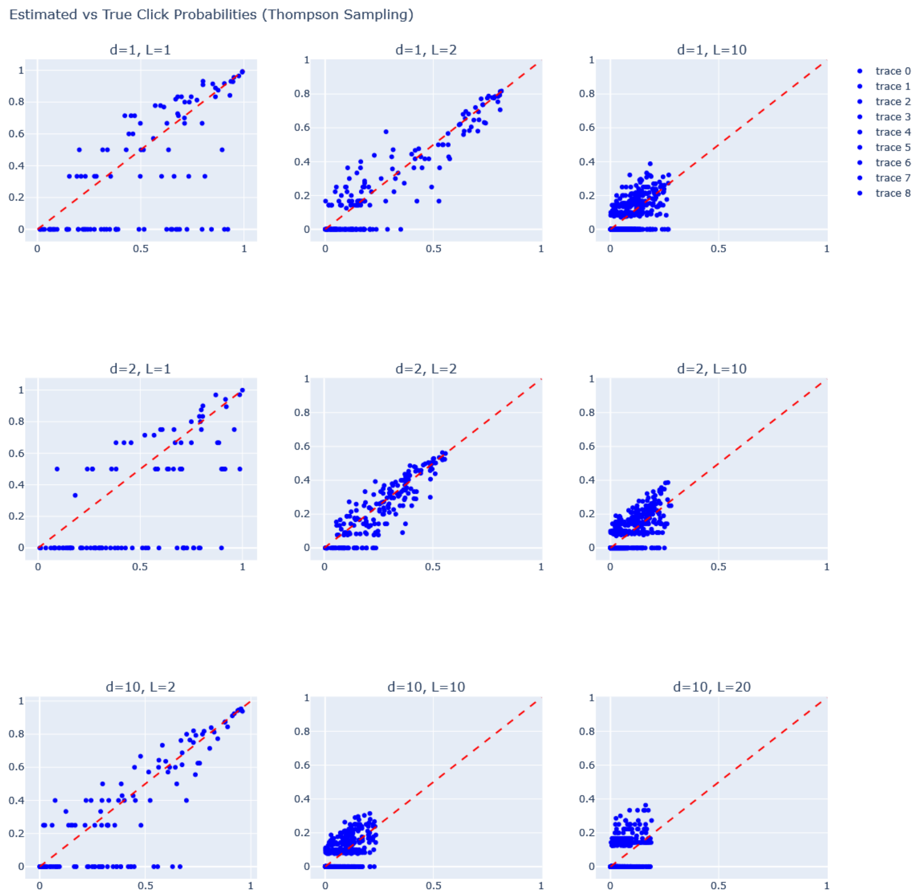

Figure 4: Estimated vs True Click Probabilities (Thompson Sampling)

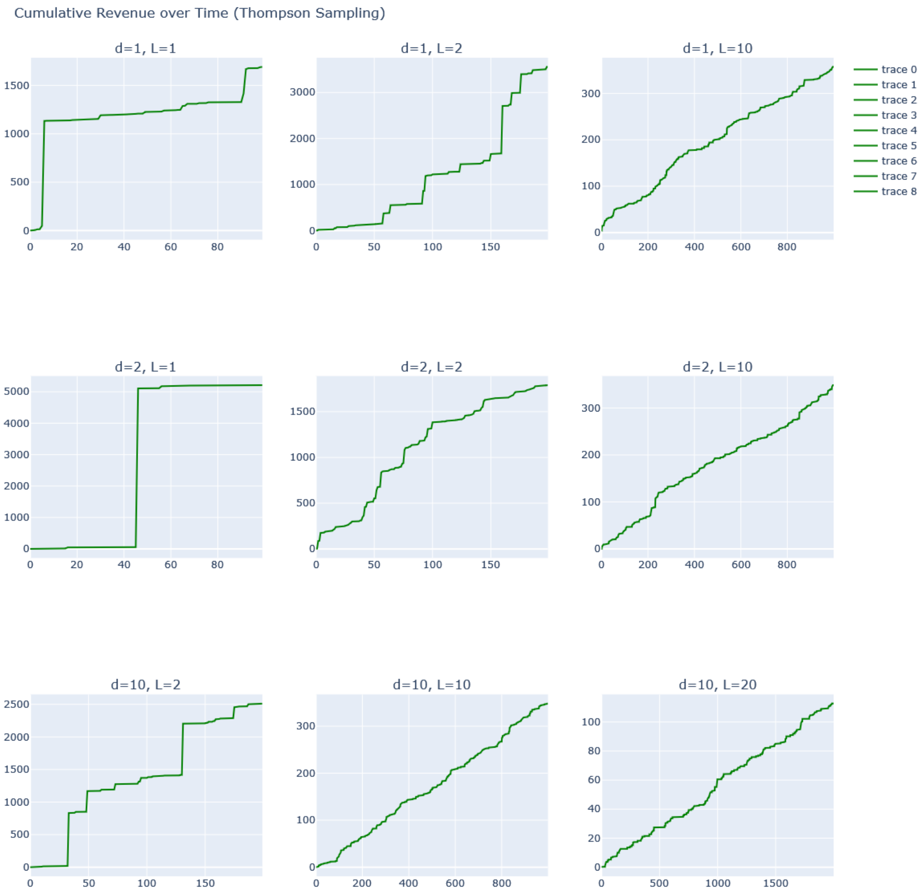

Figure 5: Cumulative Revenue over Time (Thompson Sampling)

Figure 6: Revenue per Ad (Thompson Sampling)

3.3 Performance of Adaptive AEE Strategy

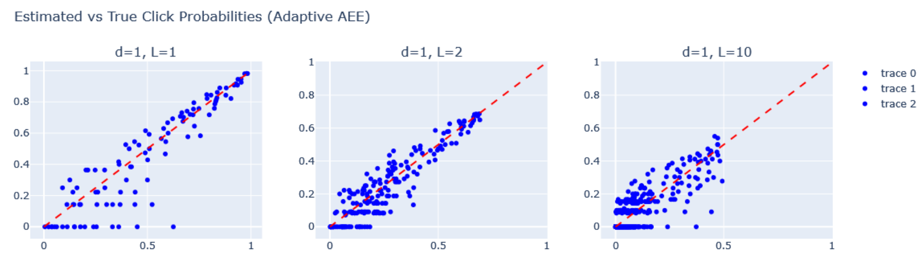

Figure 7: Estimated vs True Click Probabilities (AEE)

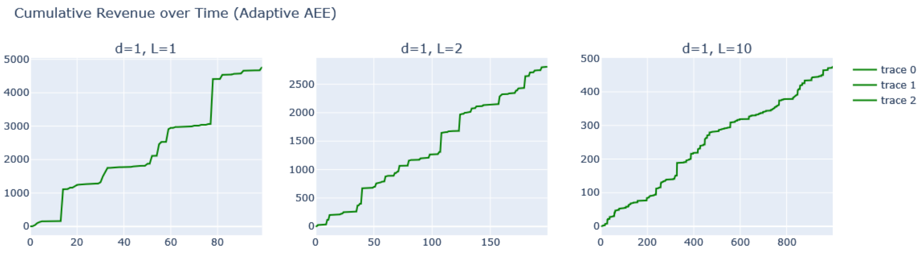

Figure 8: Cumulative Revenue over Time (AEE)

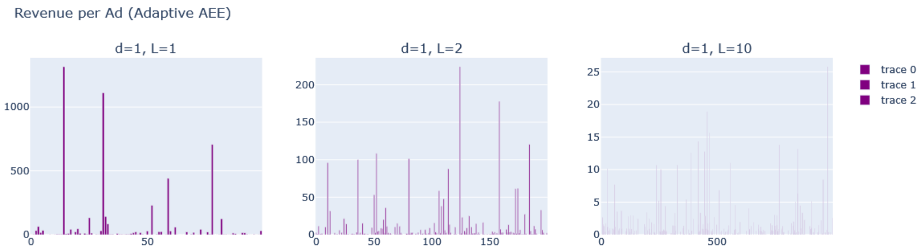

Figure 9: Revenue per Ad (AEE)

4 Comparison between UCB, Thompson Sampling, and AEE

4.1 Estimated Click Probabilities

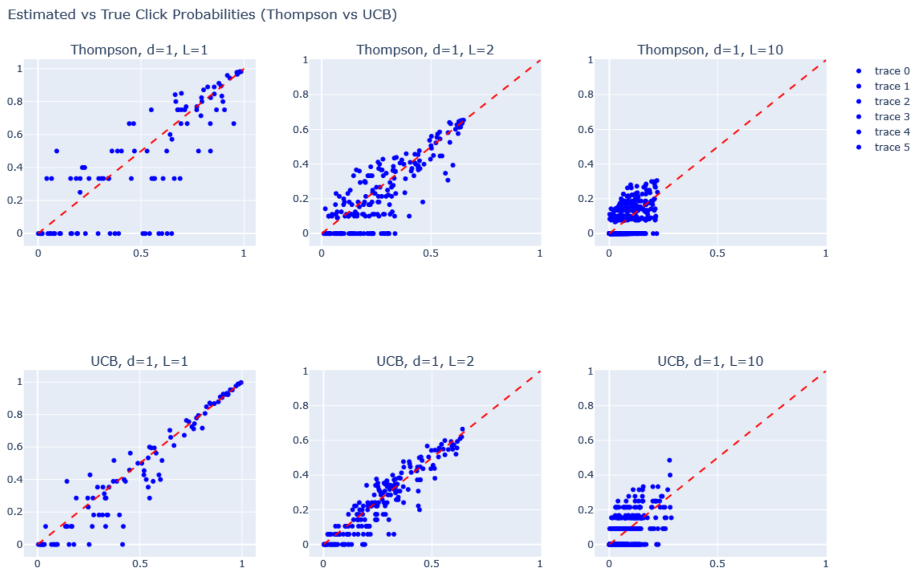

Figure 10: Comparison of Estimated vs True Click Probabilities (Thompson vs UCB vs AEE)

4.1.1 Interpretation

The scatter plots comparing estimated vs. true click probabilities provide insights into each algorithm's accuracy in estimating click probabilities.

UCB: The UCB algorithm tends to show more spread in the estimated probabilities, particularly in high-dimensional scenarios (higher L values). This spread indicates UCB's tendency to overestimate click probabilities early on due to its confidence bound, leading to potential inaccuracies in high-dimensional spaces.

Thompson Sampling: Thompson Sampling generally shows better alignment with the true click probabilities, reflecting its ability to balance exploration and exploitation more effectively. However, there is still some spread, particularly in scenarios with more sites, indicating the challenge of accurately estimating probabilities across a large number of platforms.

AEE: The AEE strategy exhibits the best alignment between estimated and true click probabilities, particularly in scenarios with higher L. This suggests that AEE's dynamic adjustment mechanism effectively balances exploration and exploitation, leading to more accurate probability estimates across different environments.

4.2 Cumulative Revenue

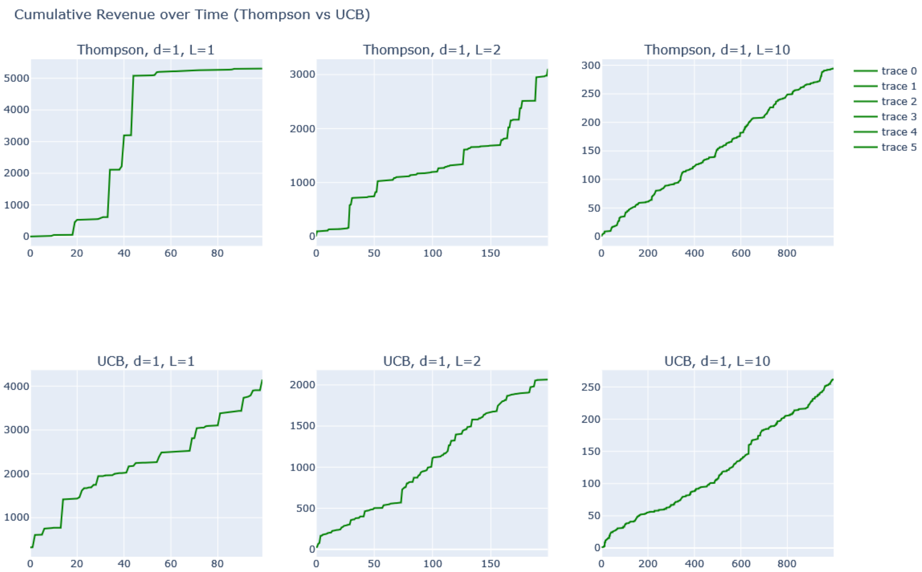

Figure 11: Comparison of Cumulative Revenue over Time (Thompson vs UCB vs AEE)

4.2.1 Interpretation

The cumulative revenue plots offer a clear comparison of how quickly and effectively each algorithm maximizes revenue over time.

UCB: UCB shows a steady increase in cumulative revenue, but it lags behind the other algorithms in scenarios with more sites. This is due to UCB's more conservative exploration approach, which may miss high-revenue opportunities early on.

Thompson Sampling: Thompson Sampling demonstrates faster revenue growth compared to UCB, particularly in the initial phases. Its ability to balance exploration and exploitation leads to quicker identification of high-revenue platforms, but its performance plateaus in complex scenarios (high L) as it struggles with the increased exploration burden.

AEE: The AEE strategy consistently outperforms both UCB and Thompson Sampling in terms of cumulative revenue growth. AEE's dynamic adjustment allows it to rapidly exploit high-revenue platforms while still exploring less known options. This leads to both higher initial revenue and sustained growth, particularly in environments with more sites.

4.3 Revenue per Ad

Figure 12: Comparison of Revenue per Ad (Thompson vs UCB vs AEE)

4.3.1 Interpretation

Analyzing the distribution of revenue per ad helps us understand how each algorithm allocates its resources across different ads.

UCB: The revenue per ad is more evenly distributed in UCB, indicating a more uniform exploration across platforms. However, this uniformity comes at the cost of lower overall revenue, as UCB does not focus as intensely on the most profitable ads.

Thompson Sampling: Thompson Sampling shows a more concentrated revenue distribution, particularly in scenarios with fewer sites. This suggests that Thompson Sampling effectively identifies and focuses on the most profitable ads, but as the number of sites increases, its ability to concentrate revenue diminishes, leading to more spread in revenue distribution.

AEE: The AEE strategy shows the most balanced approach, with revenue per ad being both concentrated on high-revenue platforms and well-distributed across others. This indicates that AEE is effectively leveraging its dynamic exploration-exploitation mechanism to maximize revenue while still ensuring sufficient exploration across all platforms.

5 Areas for Improvement

While the AEE strategy shows promising results, there are several areas where further refinement and research could enhance its performance:

5.1 Dynamic Threshold Adjustment

One challenge in the AEE strategy is setting the thresholds for when to switch between exploration and exploitation. Future work could involve developing more sophisticated methods for dynamically adjusting these thresholds based on real-time feedback, such as using machine learning models to predict when a switch is likely to be beneficial.

5.2 Scalability to Larger Systems

As the number of firms (K) and sites (L) increases, the computational complexity of the AEE strategy also increases. Developing more efficient algorithms that can handle larger-scale systems without sacrificing performance is a critical area for improvement. This could involve parallelizing certain computations or using approximation methods to reduce computational load.

5.3 Incorporating Contextual Information

The current AEE strategy, like UCB and Thompson Sampling, primarily relies on click probabilities. Incorporating additional contextual information, such as time of day, user demographics, or historical performance data, could further improve the accuracy of probability estimates and the effectiveness of exploration-exploitation decisions.

5.4 Real-World Testing and Validation

While simulation results are promising, real-world testing is crucial to validate the effectiveness of the AEE strategy. Deploying AEE in real-world advertising scenarios, with all the complexities and uncertainties that come with them, would provide valuable insights into its practical performance and areas for further refinement.

6 Impact of Parameters on Model Performance

In this section, we explore the impact of the key parameters—K (number of firms), L (number of sites), and d (dimensionality of the feature space)—on the performance of the UCB, Thompson Sampling, and AEE algorithms in the context of multi-platform ad allocation.

6.1 Impact of Number of Firms (K)

The parameter K represents the number of firms competing to advertise across the available sites. As K increases, the competition for ad space becomes more intense, leading to several implications:

- **Increased Complexity:** With more firms, the model must handle a larger number of interactions between firms and sites. This increases the computational complexity, especially in high-dimensional spaces.

- **Revenue Distribution:** Higher K typically leads to a more diverse distribution of revenue across firms, as seen in the revenue per ad plots. In such cases, firms with more effective targeting (higher click probabilities) tend to dominate the revenue share.

- **Exploration vs. Exploitation:** Both UCB and Thompson Sampling need to balance exploration and exploitation more carefully as K increases. The algorithms might initially explore more diverse ad placements before converging on the most profitable sites. AEE, with its dynamic adjustment, could potentially handle higher K values better by adapting its strategy based on real-time performance feedback.

6.2 Impact of Number of Sites (L)

The parameter L represents the number of sites (or platforms) available for advertising. Increasing L has several effects on the model:

- **Exploration Challenge:** As L increases, the number of possible firm-site interactions grows exponentially. This makes the exploration phase more challenging for the algorithms, as they must evaluate more potential placements.

- **Revenue Potential:** Higher L typically increases the overall revenue potential, as there are more platforms to target. However, this also requires the algorithms to be more efficient in identifying the most lucrative sites. AEE's dynamic exploration approach may provide an advantage in higher L scenarios by effectively balancing exploration across a large number of sites.

- **Diminishing Returns:** Beyond a certain point, adding more sites may lead to diminishing returns, as the additional sites may have lower traffic (lower s_j values), contributing less to overall revenue. AEE could potentially mitigate this by dynamically reducing exploration on low-traffic sites.

6.3 Impact of Dimensionality (d)

The parameter d represents the dimensionality of the feature space, which captures the characteristics of firms and sites. The dimensionality has a significant impact on the model's performance:

- **Curse of Dimensionality:** As d increases, the feature space becomes more complex, making it harder for the algorithms to accurately estimate click probabilities. This is evident from the scatter plots of estimated vs. true probabilities, where the accuracy tends to decrease with higher d values.

- **Generalization:** Higher dimensionality can improve the model's ability to generalize across different sites, especially when the features capture meaningful patterns. However, if d is too high, it may lead to overfitting, where the model becomes too tailored to specific interactions.

- **Computational Cost:** The computational cost increases with higher d, as both UCB and Thompson Sampling need to perform more complex calculations to account for the increased number of features. AEE may face similar challenges, but its adaptive nature could potentially allow it to prioritize the most relevant dimensions dynamically.

6.4 Interplay Between K, L, and d

The interaction between K, L, and d is crucial in determining the overall performance of the model. Some key observations include:

- **High K and High L:** In scenarios with many firms and sites, both algorithms face significant challenges in exploration. UCB may struggle more due to its reliance on confidence bounds, which become less effective in high-dimensional spaces. Thompson Sampling, with its Bayesian approach, may handle this complexity better, but at the cost of increased computational overhead. AEE, with its dynamic adjustments, might offer a balanced solution, but its performance needs to be tested in such complex scenarios.
- **High d with Low K or L:** When the dimensionality is high but the number of firms or sites is low, the algorithms might overfit to the limited data available. This is especially problematic for UCB, which may become too confident in its estimates, leading to suboptimal decisions. AEE could potentially avoid overfitting by dynamically adjusting its exploration based on the observed performance.
- **Balanced Scenarios:** The algorithms tend to perform best in balanced scenarios where K, L, and d are moderate. In such cases, the algorithms can effectively explore and exploit the available options without being overwhelmed by the complexity or constrained by limited choices. AEE's dynamic nature makes it particularly suited for these balanced scenarios, where it can continuously adjust to optimize performance.

7 Discussion

Discussion

The UCB algorithm tends to overestimate click probabilities early in the process due to its exploration term. As a result, it may initially favor platforms with higher uncertainty but eventually converges to more accurate estimates.

Thompson Sampling, on the other hand, balances exploration and exploitation more naturally by sampling from the posterior distribution. It typically achieves better revenue performance earlier in the process compared to UCB, as seen in the cumulative revenue plots.

The AEE strategy builds on the strengths of both UCB and Thompson Sampling by dynamically adjusting the exploration-exploitation balance based on real-time performance. This allows AEE to achieve faster revenue growth and more accurate click probability estimates, particularly in scenarios with a large number of sites.

However, AEE also introduces additional complexity in terms of setting and adjusting the exploration-exploitation thresholds. The effectiveness of AEE depends on how well these thresholds are tuned to the specific scenario, which could be a potential limitation in highly dynamic or unpredictable environments.

The results show that while both UCB and Thompson Sampling perform well in low-dimensional settings (e.g., $d = 1, L = 1$), Thompson Sampling generally outperforms UCB in more complex scenarios (higher L values), especially in terms of cumulative revenue. AEE further enhances performance in these complex scenarios by adapting its strategy in real-time, leading to superior results across all metrics.

8 Conclusion

Conclusion

In this study, we simulated the performance of UCB, Thompson Sampling, and a novel Adaptive Exploration-Exploitation (AEE) strategy in the context of multi-platform ad allocation. Our results indicate that while UCB is a robust algorithm with strong theoretical guarantees, Thompson Sampling often outperforms it in practice, particularly in more complex scenarios with higher dimensions and more platforms.

The AEE strategy, which dynamically adjusts the exploration-exploitation balance based on real-time performance feedback, offers a significant improvement over both UCB and Thompson Sampling. AEE consistently achieves higher cumulative revenue, more accurate click probability estimates, and a better balance between exploration and exploitation across different scenarios.

However, AEE's performance is contingent on the correct tuning of its adaptive thresholds, which introduces additional complexity. Future research should focus on refining these thresholds, improving scalability, and incorporating more contextual information to further enhance AEE's performance.

The choice of algorithm should consider the specific needs of the application, including the trade-off between exploration and exploitation, as well as computational efficiency. Parameters like the number of firms (K), sites (L), and dimensions (d) significantly impact the performance of these algorithms. Understanding these parameters' roles is crucial for effective implementation in real-world scenarios.