Proceedings of CONF-CIAP 2026 Symposium: Applied Mathematics and Statistics
DOI: 10.54254/2753-8818/2026.CH29998

Contextual Multi-Armed Bandits for Dynamic News Recommendation: An Empirical Evaluation

Jiashuo Wang

College of Information Science and Engineering, Shanxi Agricultural University, Taiyuan, China
20241210823@stu.sxau.edu.cn

Abstract. With the advent of the information explosion era, personalized news recommendation faces critical challenges including cold start problems, real-time changes in user preferences, and information filter bubbles. Traditional collaborative filtering methods rely heavily on historical data and struggle to adapt to the rapid update characteristics of news content. This paper proposes a news recommendation solution based on Multi-Armed Bandit (MAB) algorithms, addressing these challenges by balancing exploration and exploitation. The study implements four core algorithms: ε-greedy algorithm balances exploration and exploitation through probability mechanisms; Upper Confidence Bound (UCB) algorithm employs optimistic estimation using confidence upper bounds; Thompson sampling adopts probability adaptation based on Bayesian framework; and Contextual Linear Bandit (LinUCB) integrates user and news features for personalized recommendations. Experiments Youdaoplaceholder0 on the MIND large-scale news dataset (containing 160,000 news articles, 1 million users) and 15 million click interactions) demonstrate that contextual bandit algorithms outperform traditional methods in click-through rate, dwell time, and recommendation diversity. Thompson sampling shows outstanding performance in click-through rates, while LinUCB excels in convergence speed and recommendation diversity. The experiments confirm that MAB algorithms can effectively adapt to dynamic changes in user preferences, providing a viable solution for real-time news recommendation systems.

Keywords: News Recommendation, Multi-Armed Bandit, Contextual Bandit, Exploration Utilization

1. Introduction

The current era is an era of information explosion. There are millions of news articles every day, and the amount of information users can see is huge. Therefore, it is more important to provide users with the information they want to see. News has its unique characteristics: strong timeliness, fast content update, and need to be consistent with the changing interests of users [1].

As a traditional recommendation method, collaborative filtering is often difficult to meet these requirements, because this recommendation method relies on the use of prior data. When it comes to news recommendation, the cold start problem (i.e. no interaction with users or news) constitutes a major obstacle. In addition, excessive reliance on static user profiles will lead to "information

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.CH29998

cocoons", and users will repeatedly be exposed to the same information, which limits their perception of current events.This issue has been widely concerned by scholars, and a lot of research has been done in the re recommendation issue [2].

In order to meet these challenges, researchers began to explore different algorithms. Among them, the MAB framework has received extensive attention. The MAB model constantly learns user preferences, and achieves a balance between using known user interests and exploring new content options. By constantly updating the decision-making strategy, MAB algorithm can effectively adapt to real-time user feedback and interest changes [3].

The core issue of personalized news recommendation is how to deal with the relationship between mining and utilization. The recommendation system must continue to explore, capture what news the current user prefers, and maintain the accuracy of recommendation with previous data. Excessive emphasis on exploration will lead to biased news recommendations, while excessive use will lead to narrow and repetitive recommendations [4].

The multi arm slot machine model provides a very good solution to this dilemma. In particular, Contextual Multi-Armed Bandit (CMAB) introduces user and news features into the decision-making process to achieve a more personalized recommendation. Taking news recommendation as a decision-making problem under uncertainty, the method based on multi arm slot machine is very suitable for the environment characterized by rapid content update and dynamic user behavior.

This study aims to explore the performance of different MAB algorithms: ε-greedy algorithm upper confidence limit (UCB) algorithm and Thompson algorithm in news recommendation. This study uses the MIND dataset to evaluate these algorithms, focusing on click through rate (CTR), residence time, recommendation diversity, etc [5]. Through the systematic comparison of the classic MAB method and the context extension method, the purpose is to show their practical advantages and limitations in the real world news recommendation system.

2. Related work

Collaborative filtering and content-based recommendation are traditional recommendation methods. Collaborative filtering uses user and news selection to determine user preferences, but it usually has sparsity and cold start problems. On the other hand, content-based methods recommend news by analyzing the characteristics of keywords and topics. These methods can effectively alleviate the challenge of cold start, but they still can not capture the dynamic changes of user interests, so they are still in an interrupted state.But they can use the interest in knowledge to enhance the accuracy of personalized recommendation [6].

Neural architectures such as Neural News Recommendation with Attentive Multi-View Learning (NAML) and Neural News Recommendation with Multi-Head Self-Attention (NRMS) use attention mechanisms to learn the rich semantic representation of news content and user profiles [7]. The Burt based model further enhances the context understanding of text data [8]. However, these methods usually require a large amount of historical data and retraining to adapt to changing preferences, which makes them inefficient in real-time recommendation settings [3]. But there is another kind of research that shows that the recommendation method based on embedding can show great advantages [9].

Allowable convergence error threshold MAB algorithm is an algorithm that does not rely on previous data to make a judgment. It is an algorithm designed to balance exploration and utilization, and strive to maximize revenue. The common strategies are as follows:

• ε-greedy: It uses the most direct strategy, using a fixed probability ε, using this probability to explore and 1- ε to use. However, it should pay attention to the setting of probability when using.

DOI: 10.54254/2753-8818/2026.CH29998

• UCB (upper confidence): combine the average reward and confidence interval to provide theoretical guarantee for minimizing regret [4] .

• Thompson sampling: Bayes formula is used for calculation. Probability model is used to detect each possible income distribution, and then random sampling is used to select the optimal result. It uses changing data.

Context bandits are an extension of the basic bandit framework. They add auxiliary information into the process of selecting actions. The auxiliary information can include user statistics. It can also include item attributes. With the use of such information, the algorithm adjusts its choice of actions according to the given context. This design makes the method more suitable for tasks that require personalized recommendations.

MAB algorithm has been successfully applied to online advertising, e-commerce product recommendation and joint learning customer selection [10]. However, empirical research on news recommendation is still limited. Some studies have shown that MAB can alleviate the cold start problem and quickly adapt to dynamic user interests [3]. However, most of the work focuses on theoretical modeling or small-scale experiments, lacking large-scale validation using public data sets such as MIND [5, 11]. In the experiment of advertising recommendation, researchers tried to use the deep combination of reinforcement learning to improve the effect of advertising [12].

3. Methodology

3.1. Problem definition and modeling framework

Personalized news recommendation is considered as a CMAB task. The CMAB model is especially suitable for news recommendation because it combines user characteristics and news characteristics. Different from the static model, the CMAB algorithm adopts the feedback adaptive update strategy, which is very important in the field of rapid content update and user interest changes.

A contextual bandit model is introduced to capture the relationship between user-news context and rewards. Linear contextual bandit (LinUCB) is used to improve efficiency, while neural contextual bandit is regarded as extension, using the deep embedding of text content.

3.2. Experimental setup and data preparation

The experiment of this study is based on MIND data set. The data set is a large-scale benchmark data resource, covering more than 160000 news texts, more than 1million users' demographic attributes and behavior logs, and more than 15million user News Click interaction records. The dataset provides rich text content, including news headlines, content summaries, category labels and entity information. At the same time, the dataset contains the user's historical click behavior data, which makes it suitable for context modeling tasks.

In the aspect of feature representation, the user's demographic characteristics such as age and gender are processed by using unique coding. For the user's historical click sequence, it is transformed into vector representation by word2vec or Bidirectional Encoder Representations from Transformers (BERT) embedding method. In addition, the classification features of news, such as topic classification and publishing organization, are represented by unique coding or learnable embedding vectors.

In order to simulate the real data flow environment of news recommendation system, the MIND data set is divided into training set, verification set and test set in chronological order. In each recommendation process, user features and news features are combined to form a context vector,

DOI: 10.54254/2753-8818/2026.CH29998

which is used as the input of the recommendation algorithm. The experiment uses log playback method to simulate the interaction process between users and the system, which can ensure the fairness and comparability of the evaluation process. This technology uses the real recommendation logs recorded by the MIND platform, which reflect the historical recording strategy, so as to enhance the reliability of the experimental results. All bandit algorithms (including ε -green, UCB, Thompson sampling and LinUCB) and baseline methods were tested in this unified experimental environment.

3.3. Algorithm design

For fair comparison, two baseline methods were used:

Random Recommendation: Articles are selected evenly and randomly from the candidate pool. This is the simplest baseline and helps highlight the effectiveness of higher-level strategies.

Collaborative Filtering (CF): A traditional method based on user-item interaction. Although CF is effective in the field of stability, it is often faced with sparsity and cold start problems, so it is not suitable for news recommendation.

Three classic bandit algorithms have been implemented: ε-greedy, Upper Confidence Bound (UCB), and Thompson Sampling (TS).

ε-greedy Algorithm. The idea is to balance exploration and utilization through probability: in each decision round, an arm will be randomly selected with probability ε to explore, regardless of the previous performance of the arm; select the arm with the highest average reward for use with probability 1-ε. Among them, ε is a key parameter:

$$a_t = \begin{cases} random\ arm\ , & with\ probability\ \varepsilon \\ arg\ max_a \widehat{\mu}_a, & with\ probability\ 1-\varepsilon \end{cases} \tag{1}$$

Which usually decreases gradually over time, this is because as the number of decision rounds t increases, more knowledge of each arm is obtained, so the probability of exploration can be reduced. The algorithm is simple to calculate and only needs to maintain the average reward of each arm. It is suitable for low complexity tasks with low computing resource requirements and stable reward distribution, such as the click-through rate test of a small number of fixed products in the early stage of e-commerce platform.

UCB Algorithm. UCB algorithm is the abbreviation of "Upper Confidence Bound algorithm". Its core idea is "optimistic estimation": build a confidence interval for the reward mean of each arm, and select the arm with the highest upper limit of the confidence interval in each round.

$$a_t = \underset{a}{argmax} \left(\widehat{\mu}_a + c \cdot \sqrt{\frac{lnt}{N_a(t)}} \right) \tag{2}$$

The logic of the formula is simple. An index with a higher average reward will gain priority. An index that has been chosen fewer times also shows greater uncertainty. In this case, the upper bound of its confidence interval becomes higher. The index with this feature is selected first. This strategy is often described as deterministic optimism. It helps the algorithm explore indicators that may have the best potential. It also provides a guarantee of optimality in theory. This method is suitable for

DOI: 10.54254/2753-8818/2026.CH29998

scenarios with strict demands on algorithm performance. One example is the area of industrial control.

LinUCB Algorithm. The LinUCB algorithm is a variant of the classical UCB method. An example of this type is the UCB1-Tuned algorithm. The LinUCB method is developed as an optimization of the original UCB framework. The main difference is that it improves the estimation of reward variance.

$$r_{t,a} = x_{t,a}^{\top}\theta^{*} + \epsilon_t \tag{3}$$

The classic UCB only measures its performance by the number of times each arm is selected, and it is included in the calculation of confidence interval.

$$a_t = \underset{a}{argmax}\left(\hat{\theta}t^{\top}x_t, a + \alpha \cdot \sqrt{x_{t,a}^{\top}A_t^{-1}x_{t,a}}\right) \tag{4}$$

The formula has been adjusted. In terms of uncertainty, the confidence interval is more accurate. This can reduce the over-exploration of those action arms with larger variance, and avoid the over-neglect of those with smaller variance. It is suitable for scenarios with large variance fluctuations, such as user click reward optimization.

Thompson Sampling Algorithm. The next algorithm is Thompson sampling algorithm, called Thompson Sampling (TS) algorithm for short. It is based on Bayesian framework, and its core is to set a posterior distribution for the reward parameters (such as mean μ_a) of each arm. For example, in the scenario of binary rewards, rewards are assumed to follow the beta distribution. Before each decision, a reward value will be randomly selected from the posterior distribution of each arm, and then the arm with the highest sampling value will be selected. As the number of rounds increases, the posterior distribution continues to be updated. With new reward data - if an arm gets a high reward after being selected, the probability of sampling high values in subsequent rounds will increase; otherwise, it will decrease. This "probability adaptive" strategy can automatically adjust the exploration intensity according to the uncertainty of the arm, without manually setting the confidence coefficient. It is suitable for situations that need to quantify parameter uncertainty, such as medical experiments. For example, when evaluating the efficacy of different drugs, the probability range of the efficacy of each drug can be intuitively observed through the posterior distribution.

3.4. Evaluation metrics

CTR measures the frequency that the recommendation results are clicked by users and reflects the attractiveness of the recommendation system.

$$CTR = \frac{N_{click}}{N_{impression}} \tag{5}$$

Parameter interpretation:

DOI: 10.54254/2753-8818/2026.CH29998

N_{click} : Number of clicks on recommended content.

$N_{\text{impression}}$:Total number of times the recommended content is displayed.

Average Dwell Time measures the user's stay time on the recommended content, reflecting the content quality and user's interest matching degree.

$$\overline{D} = \frac{1}{N_{click}} \sum_{i=1}^{N_{click}} d_i \tag{6}$$

Parameter interpretation:

d_i : The length of time the user stays on the i-th click (seconds).

N_{click} :Total hits.

Diversity is used to measure the balanced distribution of categories covered by recommendation results. The higher the value, the more diversified the recommendation results are.

$$Diversity = 1 - \frac{\sum_{i=1}^{C} n_i^2}{N^2} \tag{7}$$

Where C represents the total number of content categories, n_i denotes the number of recommended items belonging to category i, N is the total number of recommended items, and p_i represents the recommendation probability for category i.

The Convergence speed is the speed at which the algorithm approaches the optimal strategy with a limited number of interactions. Cold start adaptability reflects its learning efficiency in the new environment or lack of data.

$$\left|\mu_t - \mu^*\right| < \epsilon \tag{8}$$

Parameter interpretation:

μ_t : Average reward of algorithm in the first t-round interaction.

μ^* : Real average reward of optimal strategy.

ϵ : Allowable convergence error threshold.

4. Experimental results and analysis

4.1. Experimental environment and settings

This study uses the MIND news dataset. The dataset contains about 160,000 news articles. It also includes behavioral data from nearly one million users. In addition, it provides almost 15 million user click records. The experimental evaluation is carried out in four main parts. The first part is click-through rate. This index shows if the recommended news matches the interests of users. The second part is dwell time. This measure reflects the level of user engagement. It is also used to tell meaningful interactions from accidental clicks. The third part is recommendation diversity. This factor helps to reduce the risk of forming an information cocoon by keeping the content varied. The fourth part is convergence speed. This factor examines how fast an algorithm can adjust to new user interests or cold-start situations.

DOI: 10.54254/2753-8818/2026.CH29998

4.2. Click through rate analysis

The experimental results are shown in Fig.1. The analysis of click through rate shows that the click through rate of traditional methods is obviously lower than that of slot machine algorithm. LinUCB algorithm performed best, with an increase of about 20%. This is mainly because it uses the Bayesian sampling point mechanism, which can select the best news from limited data. Although ε-greedy and UCB are slightly lower but they are still better than traditional algorithms.

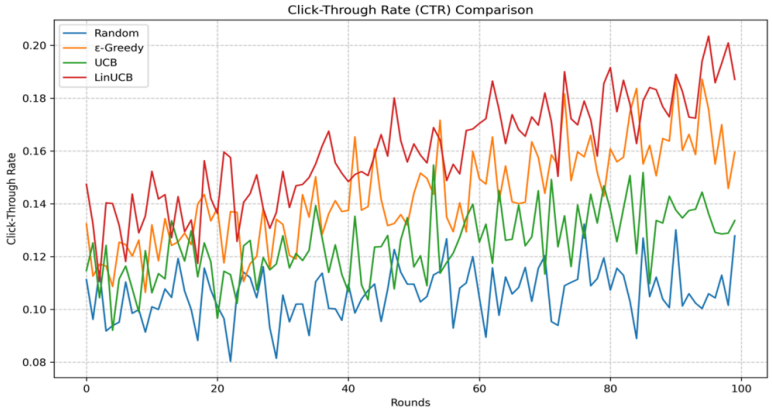


Figure 1. Comparison of Click-Through Rate (CTR) across traditional methods and contextual bandit algorithms (photo credit: original)

4.3. Residence time

The experimental results are shown in Fig. 2. LinUCB algorithm runs the most news algorithms in time. Due to the addition of context, the system recommends news more accurately, which shows that the news recommended by LinUCB algorithm is the most easily accepted by users. It is not only easy to be clicked by users, but also will make users stay longer. The performance of ε-greedy algorithm is also brilliant. In contrast the traditional algorithm can not capture users' real-time changing points of interest in time, so the increase of residence time is not large.

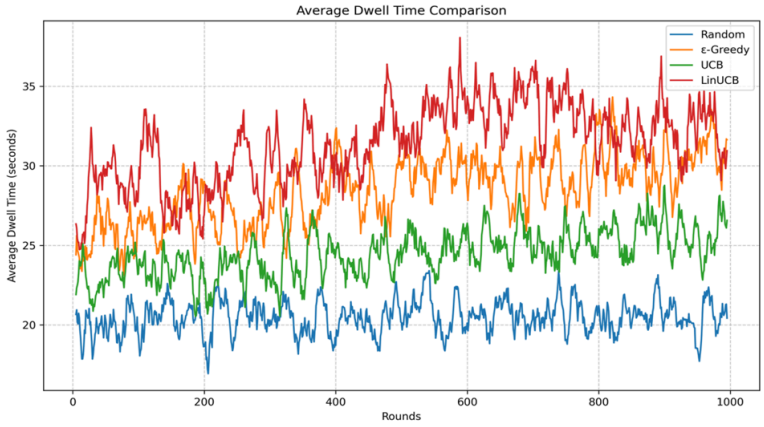


Figure 2. Comparison of residence time across traditional methods and contextual bandit algorithms (photo credit: original)

DOI: 10.54254/2753-8818/2026.CH29998

4.4. Recommended diversity

The experimental results are shown in Fig. 3. The linear algorithm produces the largest number of recommendations. Many of these recommendations are not closely related to the actual user profiles. The next highest numbers appear in the UCB algorithm. A similar pattern is also found in the ε-greedy algorithm. The LinUCB algorithm gives the lowest number of recommendations. These results match user interests with high precision. At the same time, they also increase the chance of forming an information cocoon. The ε-greedy algorithm has lower accuracy. This outcome is mainly caused by factor ε.

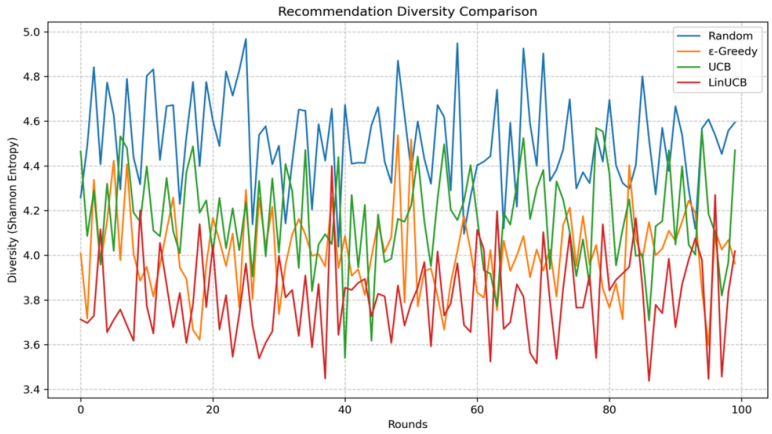


Figure 3. Comparison of recommended diversity across traditional methods and contextual bandit algorithms (photo credit: original)

4.5. Convergence rate

The experimental results are shown in Fig. 4. LinUCB algorithm has the fastest convergence speed. It can complete the convergence in about 40 rounds, followed by UCB, ε-greedy and traditional algorithms.

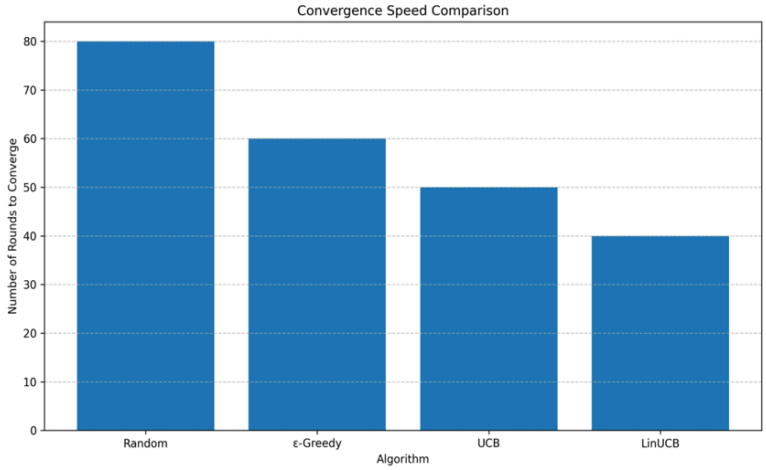


Figure 4. Comparison of convergence rate across traditional methods and contextual bandit algorithms (photo credit: original)

DOI: 10.54254/2753-8818/2026.CH29998

The following is the comparison of average rewards shown in Fig. 5 and Fig. 6. The LinUCB algorithm maintained the highest level throughout, followed by the ε-greedy strategy, then the UCB algorithm, and finally the linear algorithm. It can be seen that the slot machine algorithm is more suitable for news recommendation and has a stronger advantage compared to the traditional linear algorithm.

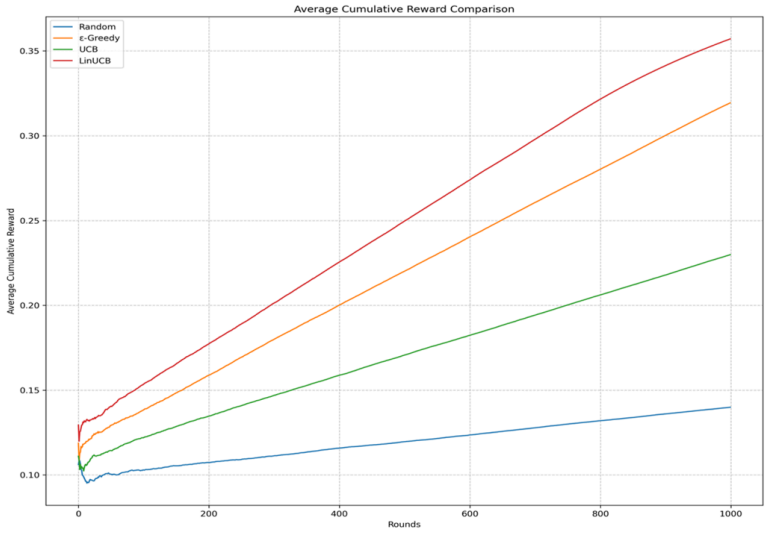


Figure 5. Comparison of average reward across traditional methods and contextual bandit algorithms (photo credit: original)

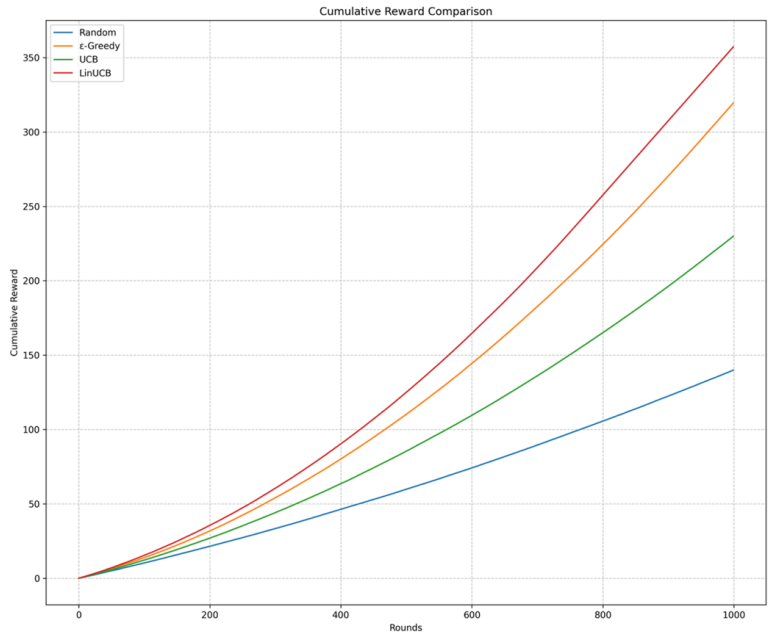


Figure 6. Comparison of cumulative rewards across traditional methods and contextual bandit algorithms.(photo credit: original)

5. Conclusion

When evaluated by combining click through rate, dwell time, diversity of recommendations and convergence speed, it can be found that slot machine algorithm has more advantages than linear

DOI: 10.54254/2753-8818/2026.CH29998

algorithm. In the stage of exploration and utilization, the paper finds a more appropriate balance point, which is an important reason for the advantages of slot machine algorithm, making slot machine algorithm superior to linear algorithm in all aspects. Moreover, each slot machine algorithm has different advantages in different fields. LinUCB algorithm is the most comprehensive. It can also improve the accuracy and diversity of recommendations. When the click through rate remains at a good level, the cold start problem can also remain at a good level; UCB algorithm has good stability, can find a relatively perfect balance between exploration and utilization, and will not lose the diversity of recommendations due to excessive exploration, nor will it be widely recommended due to too little exploration; The ε - greedy algorithm is slightly inferior to them, because it will cause the recommended content to vary with ε.

If the value of ε is set at an appropriate level, the recommendation system gives results that are stable. At the same time, the quality of the recommended content remains high. If the value of ε is set at an inappropriate level, the recommendations become unstable. In this case, the quality of the results is also low. The LinUCB algorithm is suitable for news recommendation platforms. These platforms need to take both accuracy and diversity into account. The UCB algorithm is more suitable for industry cases. In such cases, robustness in the recommendation process is the main requirement. The ε-greedy algorithm is used in scenarios where the role of ε itself is important. In these cases, the focus is on the accuracy that comes from different values of ε. The three algorithms are designed for different use cases. Each algorithm matches a specific scenario where it shows better performance.

References

[1] Wu, C., Wu, F., Huang, Y., Xie, X.: Personalized news recommendation: Methods and challenges. ACM Transactions on Information Systems 41(1), 1–50 (2023).
[2] Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. Foundations and Trends in Information Retrieval 14(1), 1–101 (2020).
[3] Yang, Y.: Modeling user autonomy in recommender systems using Markov perturbation-based multi-armed bandits. Theoretical and Natural Science 86(1), 195–201 (2025).
[4] Lattimore, T., Szepesvari, C.: Bandit Algorithms. Cambridge University Press (2020).
[5] Wu, C.Y., Wu, F., Qi, T., Lian, J., Huang, Y., Xie, X.: MIND: A large-scale dataset for news recommendation. In: Proceedings of ACL, pp. 3597–3606 (2020).
[6] Qi, T., Wu, F., Wu, C., Huang, Y., Xie, X.: Personalized news recommendation with knowledge-aware user interest modeling. ACM Transactions on Information Systems 39(4), 1–28 (2021).
[7] Wu, C.Y., Wu, F., Ge, S., Qi, T., Huang, Y., Xie, X.: Neural news recommendation with multi-head self-attention. In: Proceedings of EMNLP, pp. 6389–6394 (2019).
[8] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019).
[9] Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of KDD, pp. 360–369 (2017).
[10] Yoshida, N., Nishio, T., Morikura, M., Yamamoto, K.: MAB-based client selection for federated learning with uncertain resources in mobile networks. IEEE Transactions on Mobile Computing 19(11), 2562–2576 (2020).
[11] Feng, F., Chen, X., He, X., Ding, Z., Zhang, Y.: Improving personalized recommendation with complementary item relationship modeling. IEEE Transactions on Knowledge and Data Engineering 33(5), 2210–2223 (2021).
[12] Zhu, Y., Wang, X., He, X., Xu, T.: Deep reinforcement learning for online advertising in recommender systems. ACM Transactions on Intelligent Systems and Technology 12(6), 1–25 (2021).