Proceedings of CONF-CDS 2026 Symposium: Data-Centric AI Security: Securing Models, Learning Agents, and Autonomous Systems
DOI: 10.54254/2755-2721/2026.CH35582

Multi-Armed Bandit Algorithms: Learning from Experience

Ziyue Zhou

International Business School, Xi'an Jiaotong-Liverpool University, Suzhou, China
zzy200512282024@163.com

Abstract. Consider a scenario where a decision-maker faces a row of slot machines, each offering unknown and varying payout rates. The objective is to maximize cumulative rewards, yet each action simultaneously provides new information. This problem—balancing the exploration of new options against the exploitation of known rewarding ones—is termed the multi-armed bandit problem. It has evolved from a simple gambling question into a fundamental tool for modern computer systems. This paper looks at three main ways to solve this problem: Explore-Then-Commit, Upper Confidence Bound, and Thompson Sampling. Through careful testing, Thompson Sampling stands out as the best performer, cutting total regret down to 0.60 where UCB reaches 3.63. It also handles delays well—when feedback comes 1000 steps late, Thompson Sampling's lead over UCB grows to nearly 4 times. The paper shows where these methods are used in real life: online ads, where they improve click rates by about 11%; movie and music suggestions, where they help new users find content they like; and medical trials, where they can put over 80% of patients on better treatments. The paper ends with a look at new research areas, including methods that adapt to changing conditions and handle complex choices.

Keywords: multi-armed bandit, exploration-exploitation trade-off, Thompson Sampling, Upper Confidence Bound, regret minimization

1. Introduction

The multi-armed bandit problem was first proposed by Herbert Robbins in 1952, describing a scenario where a decision-maker selects among multiple slot machines with unknown reward probabilities to maximize cumulative rewards [1]. This problem encapsulates the fundamental exploration-exploitation dilemma, which is ubiquitous in real-world decision-making scenarios, including route selection, learning material choice, and product development decision-making [2].

In 1985, Tze Leung Lai and Herbert Robbins proved the logarithmic regret lower bound, establishing the theoretical performance limit for bandit algorithms [3]. Subsequently, the UCB1 algorithm proposed by Peter Auer et al. in 2002 realized finite-time performance guarantees, and Thompson Sampling, originally proposed in 1933, was rediscovered and widely applied [4-6]. These algorithms have become core solutions to the multi-armed bandit problem.

The multi-armed bandit problem provides a unified theoretical framework for sequential decision-making under uncertainty. Resolving this problem can optimize decision-making efficiency in resource allocation, user interaction, and clinical trials, reduce unnecessary trial costs, improve

© 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: 10.54254/2755-2721/2026.CH35582

cumulative returns, and offer methodological support for the development of adaptive intelligent systems.

This paper centers on the exploration-exploitation trade-off in sequential decision-making under uncertainty. It evaluates three classical algorithms—Explore-Then-Commit, Upper Confidence Bound, and Thompson Sampling—through comparative experiments and analyzes their performance in terms of regret minimization and robustness to delayed feedback. The study demonstrates the practical value of bandit algorithms in real-world applications.

2. The basics: setting up the problem

2.1. The math setup

Picture K slot machines in a row, each labeled i = 1, 2, ..., K. Each machine i has a reward pattern with an unknown average μ_i. The game goes in steps t = 1, 2, ..., T. At each step, you pick a machine a_t and get a reward X_t from that machine's pattern [7].

Consider K slot machines, each with an unknown expected reward μ_i [8]. At each time step t, the decision-maker selects a machine a_t and obtains a random reward X_t. The optimal expected reward is defined as $\mu^* = \max_i \mu_i$, and the reward gap of machine i is $\Delta_i = \mu^* - \mu_i$.

2.2. Measuring success: regret

Total regret R_T measures the performance loss of an algorithm relative to the optimal strategy, calculated as [7]:

$$R_T = T \cdot \mu^* - \sum_{t=1}^{T} \mu_{a_t} \tag{1}$$

This can be rewritten to show how each machine contributes. Let $T_i(T)$ count how many times machine i was picked through step T. Then:

$$R_T = \sum_{i:\mu_i<\mu^*} T_i(T) \cdot (\mu^* - \mu_i) \tag{2}$$

The optimal theoretical performance of bandit algorithms is logarithmic regret:

$$R_T = O(\log T) \tag{3}$$

This means the total cost of mistakes grows slowly with time [2].

2.3. The core tension: explore or exploit

The exploration-exploitation trade-off is the core of the bandit problem. Exploitation refers to selecting options with known high rewards, while exploration refers to trying unknown options to obtain more information. This trade-off widely exists in application scenarios such as recommendation systems and clinical trials, directly affecting decision-making effectiveness and ethical rationality [9, 10].

DOI: 10.54254/2755-2721/2026.CH35582

3. Three main methods

3.1. Explore-Then-Commit: the simple approach

Explore-Then-Commit (ETC) divides the decision process into two independent stages. In the exploration stage, each machine is tested m times in advance to estimate the average reward; in the commitment stage, the machine with the highest estimated reward is selected for all subsequent steps. ETC has the advantage of simple logic and easy implementation, but it requires pre-setting the number of explorations, lacks adaptive adjustment capability, and easily causes waste of exploration resources or wrong permanent selection due to insufficient exploration [11].

3.2. Upper Confidence Bound: optimism in action

The Upper Confidence Bound (UCB) algorithm integrates exploration and exploitation in each decision by constructing an upper confidence bound for the expected reward of each machine [3, 4]. The UCB1 algorithm selects the machine with the highest upper confidence bound at each step, balancing empirical rewards and uncertainty. UCB1 has clear finite-time theoretical guarantees and does not require manual parameter tuning, but its deterministic selection strategy leads to over-exploration when multiple machines have similar rewards, resulting in higher actual regret [4].

3.3. Thompson Sampling: the probabilistic approach

Thompson Sampling is a Bayesian-based probabilistic algorithm that maintains posterior probability distributions of the expected rewards for each machine [6]. At each step, the algorithm samples from the posterior distributions and selects the machine with the highest sampled value. This random sampling strategy naturally balances exploration and exploitation, avoids the oscillation problem of deterministic algorithms, and has strong adaptability to delayed feedback and noise interference, showing optimal practical performance [12, 13].

4. Comparing the methods

4.1. Test setup

To see how these methods perform, researchers run simulation tests. A common setup uses 5 machines with reward averages from a $\mathrm{Beta}(8,8)$ distribution. The methods run for 1000 steps, with results averaged over 100 runs to reduce random noise [14].

4.2. Results over time

Table 1. Regret growth over time

Step	ETC	UCB1	Thompson Sampling
0	0.99	0.00	0.00
50	7.22	13.47	6.00
100	16.30	15.90	6.99
150	16.77	14.92	7.59
200	15.27	14.52	7.71

DOI: 10.54254/2755-2721/2026.CH35582

Table 1. (continued)

250	15.52	15.21	8.32
300	17.58	16.59	8.72
350	17.02	19.78	9.46
400	16.03	20.46	9.49
450	17.29	19.91	9.09
500	16.54	18.26	9.92
550	16.78	18.96	9.90
600	17.74	19.32	9.99
650	15.84	21.05	10.55
700	16.28	23.45	10.84
750	17.69	22.05	10.96
800	17.49	21.19	10.58
850	19.06	19.04	10.88
900	18.09	20.04	11.20
950	17.84	22.87	11.53
1000	20.97	24.92	11.22

Table 1 reveals that ETC has a sharp increase in regret during the exploration stage and remains at a high level. UCB1 has a steady growth of regret but is always higher than Thompson Sampling. Thompson Sampling maintains the lowest regret throughout the process, and its total regret is about half of UCB1 and one-third of ETC after 1000 steps [14].

4.3. Handling delayed feedback

Real systems often don't give immediate feedback. Table 2 tests how the methods handle this.

Table 2. Performance with delayed feedback

Delay (steps)	UCB Total Regret	Thompson Sampling Total Regret	UCB/TS Ratio
1 (instant)	24,145	9,105	2.65
10	25,662	9,049	2.84
100	37,141	11,550	3.22
1000(long)	226,220	59,256	3.82

Under delayed feedback conditions, both algorithms experience performance degradation, but Thompson Sampling degrades more slowly. When the feedback delay reaches 1000 steps, the regret ratio of UCB to Thompson Sampling rises to 3.82, which verifies that Thompson Sampling has stronger robustness to delayed feedback [6].

DOI: 10.54254/2755-2721/2026.CH35582

5. Real uses

5.1. Online advertising: getting more clicks

Online ad systems run billions of auctions daily. Each ad version—different headlines, images, and calls-to-action—is a machine, and user clicks are the rewards. Standard A/B testing splits traffic equally regardless of performance. If version A gets 2% clicks and B gets 3%, a test with 10,000 samples each will find B is better—but only after wasting 5,000 impressions on worse A. This fixed split ignores growing evidence, sacrificing revenue for statistical purity.

Bandit methods offer a flexible alternative. As data comes in, they shift toward better versions, learning and earning together. A study of an "Augmented Two-Stage Bandit" in live ad systems showed this clearly: during cold-start periods with little data, the bandit method improved click rates by 10.95% over baselines. This gain was strongest in the first two days—when first impressions matter most.

The business impact goes beyond percentages. In large systems, even small click rate improvements mean millions in added revenue. Bandit methods also cut opportunity cost: instead of locking half the traffic into worse versions for weeks, they optimize continuously, showing most users high-performing ads throughout.

Thompson Sampling is especially common in ad platforms due to its strong handling of delayed feedback—clicks often don't register immediately. While UCB's fixed indices get stale when feedback lags, Thompson Sampling's probabilistic approach stays effective, as shown in Table 2's tests [15].

5.2. Recommendation systems: helping new users

Every recommendation system faces the "cold start" problem with new users. When someone first joins Netflix or Spotify, the system knows nothing about them—no viewing history, no ratings. Standard methods that find "users like you" fail because there are no similar users in the database yet. Research shows users who don't see relevant content early are much more likely to quit.

Bandit methods turn this problem into an opportunity. By treating each movie or song as a machine and user engagement as rewards, they enable immediate personalization starting from the first interaction. For a new user, the method explores broadly, trying varied content to see reactions. After just a few interactions—maybe 5-10 clicks—patterns emerge, and the method shifts to exploiting, suggesting content similar to what worked.

The speed advantage is large and important. Where standard methods might need hundreds of interactions to build a profile, bandit methods start personalizing after just a few. This matters because user attention is short: a service that suggests a hit show on day one builds loyalty; one that shows random content for a month loses users to competitors.

Contextual extensions boost this further. By including demographics, device type, time of day, or browsing context, contextual bandits can generalize across similar users. A young city user on Friday evening might get different suggestions than a rural retiree on Tuesday morning, even with identical histories. The LinUCB method, which models rewards as linear functions of context, works well for news recommendations where interests shift fast with current events.

Tests in recommendation settings confirm Thompson Sampling's edge. In one matrix factorization study, it got a regret of 0.60 vs. ε-greedy's 1.59 and UCB's 3.63—showing both better performance and more stability. On ranking quality (NDCG), it scored 0.959 vs. UCB's 0.691, meaning it not only picked better items but also ranked them better [9].

DOI: 10.54254/2755-2721/2026.CH35582

5.3. Medical trials: ethics and efficiency

No application has higher stakes than medical trials, where method choices directly affect patient survival. The standard approach—fixed 50-50 randomization—has statistical benefits but ethical costs that grow uncomfortable as evidence accumulates. If a new treatment shows clear superiority mid-trial, continuing to assign patients to the worse treatment seems morally wrong, even if statistically required.

Adaptive trial designs using bandit methods offer an ethical alternative. By treating treatments as machines and patient outcomes as rewards, these designs adjust assignment probabilities based on accumulating evidence. Early on, when unsure, assignment stays roughly balanced to gather data. As evidence builds, probability shifts toward better treatments—without hurting the trial's ability to find significant results.

Simulation studies show the scale of possible gains. In trials comparing a control with a 50% success rate vs a treatment with a 70% success rate, bandit methods can assign over 80% of patients to the better treatment while keeping statistical power above 80%.

The benefits go beyond patient welfare to trial efficiency. By focusing patients on better treatments, bandit methods can reduce the total sample size needed, potentially cutting trial time and costs. The "controlled Gittins index" approach, which keeps some patients for control while adaptively assigning the rest to treatments, shows promise for balancing these goals.

Regulators have recognized these benefits. The FDA has issued guidance supporting adaptive designs, noting they "may be useful in clinical trials" when done properly. This has sped up adoption, with bandit designs now in trials for cancer, heart disease, and infectious diseases [10].

6. Conclusion

The multi-armed bandit problem centered on the exploration-exploitation trade-off is a core issue in sequential decision-making under uncertainty. Among the three classical algorithms, Explore-Then-Commit is simple but inflexible, UCB has solid theoretical support but poor practical performance, and Thompson Sampling has the lowest regret and strong robustness to delayed feedback, showing optimal comprehensive performance.

Current research has limitations. Most algorithms rely on the stationary reward distribution assumption, which is difficult to adapt to dynamic real scenarios. Contextual and combinatorial bandit algorithms face challenges in computational complexity and model generalization. The theoretical analysis of Thompson Sampling needs to be improved to narrow the gap between theory and practice.

Future research should focus on non-stationary bandit algorithms adaptive to dynamic environments, efficient contextual bandit algorithms integrating rich feature information, and scalable combinatorial bandit algorithms for complex decision combinations, so as to expand the application scope and performance of bandit methods and provide support for intelligent decision-making systems in complex uncertain environments.

References

[1] Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society, 58*(5), 527–535.

[2] Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing, 32*(1), 48–77.

DOI: 10.54254/2755-2721/2026.CH35582

[3] Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6*(1), 4–22.
[4] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning, 47*(2–3), 235–256.
[5] Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika, 25*(3–4), 285–294.
[6] Chapelle, O., & Li, L. (2011). An empirical evaluation of Thompson sampling. *Advances in Neural Information Processing Systems, 24*, 2249–2257.
[7] Kuleshov, V., & Precup, D. (2014). Algorithms for the multi-armed bandit problem. *Journal of Machine Learning Research*, 1–49.
[8] Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning, 5*(1), 1–122.
[9] Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (pp. 661–670).
[10] Villar, S. S., Bowden, J., & Wason, J. (2015). Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statistical Science, 30*(2), 199–215.
[11] Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning, 12*(1–2), 1–286.
[12] Russo, D., & Van Roy, B. (2014). Learning to optimize via information-directed sampling.
[13] (2019). Thompson sampling based methods for reinforcement learning. *Columbia University Tutorial*.
[14] (2026). A comparative analysis of cumulative regret based on multi-armed bandit algorithms. *American Journal of Science and Technology*.
[15] Xu, S. (2021). BanditMF: Multi-armed bandit based matrix factorization recommender system. *ArXiv*, abs/2106.10898.