Proceedings of CONF-CDS 2026 Symposium: Data-Centric AI Security: Securing Models, Learning Agents, and Autonomous Systems
DOI: 10.54254/2755-2721/2026.CH33252

A Comparison of ε-Greedy and Thompson Sampling in Multi-Armed Bandit Problems

Ruihan Yu

Haide College, Ocean University of China, Qingdao, China
23220003103@stu.ouc.edu.cn

Abstract. The multi-armed bandit problem is a fundamental framework for sequential decision-making under uncertainty, widely applied in online advertising, recommendation systems, and clinical trials. Balancing exploration and exploitation is crucial for maximizing cumulative rewards. This paper compares two popular algorithms: the ε-greedy strategy and Thompson Sampling. Through a literature review and theoretical analysis, this paper examines their exploration mechanisms, learning efficiency, and practical applicability. Although the ε-greedy approach is straightforward and computationally effective, its reliance on fixed random exploration may result in less than ideal performance. In contrast, Thompson Sampling uses Bayesian inference to adaptively explore based on posterior uncertainty, achieving a more effective trade-off. Empirical results from Bernoulli bandit simulations show that Thompson Sampling accumulates significantly lower cumulative regret over time. A side-by-side comparison table highlights key differences in adaptivity, computational cost, and asymptotic regret. This study also discusses recent advances and provides guidance for algorithm selection in real-world applications.

Keywords: Multi-Armed Bandit, Thompson Sampling, ε-Greedy, Exploration-Exploitation, Bayesian Learning

1. Introduction

Sequential decision-making under uncertainty is a critical problem in many real-world fields, including online advertising, recommendation systems, and clinical trials. The multi-armed bandit problem provides a formal framework for such scenarios, where a learner repeatedly chooses among k actions (arms) and receives a reward drawn from an unknown stationary distribution associated with each action. The goal is to maximize the expected cumulative reward over a given time horizon. A key challenge is the exploration–exploitation trade-off: exploitation selects the arm with the highest estimated reward based on current knowledge, while exploration tries less-selected arms to improve estimates and potentially discover better options. Achieving the right balance is essential for long-term performance [1].

Previous studies have proposed numerous algorithms to address this trade-off. Among them, the ε-greedy strategy and Thompson Sampling are two widely used approaches. The ε-greedy algorithm selects the arm with the highest estimated reward with probability 1-ε and a random arm with probability ε [1]. Thompson Sampling, a Bayesian method, samples from the posterior distribution

DOI: 10.54254/2755-2721/2026.CH33252

of each arm's reward parameter and selects the arm with the highest sampled value [2]. Later research has shown that each algorithm has its own advantages in different scenarios [3]. Recent surveys have highlighted the importance of comparative algorithm evaluation, noting that no single algorithm dominates all settings [4].

This paper compares these two popular algorithms—Thompson Sampling and the ε-greedy strategy. It reviews existing literature on bandit algorithms and analyzes their underlying mechanisms. This study aims to help practitioners understand the differences between these algorithms and choose the right one for real-world problems.

2. Multi-armed bandit problem

In the multi-armed bandit problem, there are k actions (arms). At each time step t, the learner selects an arm A_t and receives a reward R_t drawn from a stationary probability distribution specific to that arm. The expected reward of arm i is $\mu_i = E[R_t|A_t = i]$, which is unknown to the learner. The objective is to maximize the total expected reward over T steps, or equivalently to minimize the cumulative regret defined as $L_T = T\mu* - \Sigma_{t=1}^{T} E[R_t]$, where $\mu* = \max_i \mu_i$ is the optimal expected reward [5]. The main dilemma is that the learner must decide whether to exploit the arm that currently appears best or to explore other arms that may yield higher rewards in the long run. If the learner focuses too much on exploitation, it may miss better actions. Excessive exploration, however, reduces short-term rewards. A successful algorithm must balance these two aspects.

3. Greedy algorithms

3.1. Pure greedy strategy

The pure greedy method selects the action with the highest estimated value at each time step:

$$A_t = \arg\max_i \hat{Q}_t(i) \tag{1}$$

where $\hat{Q}_t(i) = \frac{1}{N_t(i)} \sum_{j=1}^{t} r_j 1(A_j = i)$ is the sample mean reward of arm i up to time t, and $N_t(i)$ is the number of times arm i has been selected. Although this approach is simple and computationally efficient (O(k) per step, where k is the number of arms), it suffers from a fundamental flaw: insufficient exploration. Because early observations may be misleading due to sampling noise, the greedy algorithm can permanently commit to a suboptimal arm. For example, consider two arms with true means 0.6 and 0.4. If the first two pulls of the suboptimal arm yield rewards 1 and 1 (giving an estimated mean of 1.0), while the optimal arm returns 0 and 0 (estimated 0.0), the greedy method will never revisit the optimal arm. When multiple actions have the same estimated value, ties are broken arbitrarily (e.g., uniformly at random) [1]. This naive exploitation leads to linear regret in many practical scenarios.

3.2. ε-Greedy strategy

To address the exploration shortfall, the ε-greedy algorithm introduces forced random exploration. With probability 1-ε, it exploits by selecting $A_t = \arg\max_i \hat{Q}_t(i)$; with probability ε, it explores

DOI: 10.54254/2755-2721/2026.CH33252

by choosing uniformly at random from all k arms. The decision rule can be written as:

$$A_t = \begin{cases} \arg\max_i \widehat{Q}_t(i) & \text{with probability } 1-\epsilon \\ \text{Uniform}\{1,\ldots,k\} & \text{with probability } \epsilon \end{cases} \tag{2}$$

This simple modification allows the learner to occasionally try alternative actions, reducing the risk of being trapped in suboptimal decisions. ε-greedy provides a more balanced exploration-exploitation trade-off and generally achieves better long-term performance [1]. However, a fixed ε leads to linear regret asymptotically, because the algorithm continues to explore with constant probability even after the optimal arm has been identified. An improvement is to decay ε over time, e.g., $\epsilon_t = 1/t$, achieving logarithmic regret in stationary environments. Nonetheless, the exploration mechanism remains random and does not leverage uncertainty information across arms, which Thompson Sampling overcomes via Bayesian posterior sampling.

From a computational perspective, ε-greedy requires only O(k) operations per step (to compute the current maximum), which is often negligible even for large k. This simplicity makes it attractive for real-time systems with tight latency constraints.

4. Thompson sampling

4.1. Bayesian formulation

Thompson Sampling treats the bandit problem from a Bayesian perspective. Instead of estimating each arm's reward with a point estimate, it maintains a posterior distribution over the unknown parameter θ_i (e.g., the success probability for Bernoulli rewards). At each time step t, the algorithm draws a sample $\tilde{\theta}_i^{(t)}$ from the posterior of each arm i, then selects $A_t = \arg\max_i \tilde{\theta}_i^{(t)}$. This randomized selection rule has a natural interpretation: the probability of selecting arm i equals the posterior probability that i is optimal [2]. Compared with ε-greedy, Thompson Sampling does not rely on a fixed exploration rate; instead, exploration emerges from posterior uncertainty.

4.2. Bernoulli bandit implementation

For Bernoulli rewards (each pull yields either 0 or 1), the conjugate prior is the Beta distribution. Arm i is associated with parameters (α_i, β_i) , where α_i can be interpreted as the number of observed successes plus a prior count, and β_i the number of failures plus a prior count. A common choice is the uniform prior $\text{Beta}(1,1)$. The update rule after observing reward $r_t \in 0,1$ is:

$$(\alpha_{A_t}, \beta_{A_t}) \leftarrow \begin{cases} (\alpha_{A_t}+1, \beta_{A_t}) & \text{if } r_t = 1 \\ (\alpha_{A_t}, \beta_{A_t}+1) & \text{if } r_t = 0 \end{cases} \tag{3}$$

The posterior mean $\mathbb{E}[\theta_i] = \alpha_i/(\alpha_i+\beta_i)$ gives an estimate of the arm's true success probability, while the variance $\frac{\alpha_i\beta_i}{(\alpha_i+\beta_i)^2(\alpha_i+\beta_i+1)}$ captures remaining uncertainty. A comprehensive treatment of Thompson Sampling and its variants can be found in [6].

DOI: 10.54254/2755-2721/2026.CH33252

4.3. Uncertainty-driven exploration

The key insight is that exploration is driven by posterior variance. When an arm has been tried only a few times, its Beta distribution is wide (high variance), so the sampled value $\tilde{\theta}_i$ can sometimes be large even if the posterior mean is low. This creates a natural tendency to try uncertain arms. As more data accumulates, the distribution concentrates around the true mean, and the algorithm exploits more reliably. This mechanism is fundamentally different from ε-greedy, which explores blindly with probability ε regardless of how much is already known about an arm.

Despite its elegance, Thompson Sampling has practical drawbacks. First, maintaining posterior distributions requires storing parameters for each arm (O(k) memory, which is acceptable, but sampling from non-conjugate families can be computationally heavy). Second, the algorithm requires specifying a prior; while uninformative priors (e.g., $\mathrm{Beta}(1,1)$) work well in many cases, a misspecified prior can hurt performance, especially in the small-sample regime. Third, for reward models beyond the exponential family (e.g., heavy-tailed or censored rewards), exact posterior sampling may be infeasible, and approximate methods (e.g., Laplace approximation or MCMC) add complexity. These limitations explain why ε-greedy remains applealing in many production systems despite its theoretical suboptimality.

5. Comparison of algorithms

5.1. Algorithm comparison summary

Table 1 provides a side-by-side comparison of pure greedy, ε-greedy, and Thompson Sampling along five key dimensions. The pure greedy strategy serves as a baseline for understanding how exploration mechanisms affect performance.

Table 1. Comparison of greedy, ε-greedy, and thompson sampling

Dimension	Pure Greedy	ε-Greedy	Thompson Sampling
Exploration mechanism	None (only exploits)	Random with fixed prob ε	Bayesian posterior sampling
Exploration adaptivity	—	No (fixed rate)	Yes (uncertainty-driven)
Prior knowledge required	No	No	Yes
Per-step computation	O(k)	O(k)	O(k)
Asymptotic regret	Linear	Linear (for fixed ε)	Logarithmic (optimal)

As summarized in the table, Thompson Sampling differs from ε-greedy primarily in its adaptive exploration mechanism and asymptotic regret guarantees. While ε-greedy explores blindly at a constant rate, Thompson Sampling automatically reduces exploration as uncertainty decreases. This adaptivity is the key to its superior long-term performance.

5.2. Mechanistic differences

The ε-greedy strategy separates exploration and exploitation into distinct, random actions. With probability ε, it ignores all learned information and picks uniformly at random. This approach is easy to implement and debug, but it can lead to two inefficiencies: exploring arms that are already known to be suboptimal, and failing to explore arms that have high uncertainty but moderate estimates. Thompson Sampling, by contrast, integrates exploration into the selection process

DOI: 10.54254/2755-2721/2026.CH33252

probabilistically. Every pull is based on a sample from the posterior, so even when the algorithm "exploits", it does so with a degree of uncertainty. As a result, Thompson Sampling typically identifies the optimal arm faster [3].

5.3. Regret and sample efficiency

From a regret perspective, a fixed ε leads to linear regret because the algorithm never stops exploring at rate ε. In practice, practitioners often decay ε over time (e.g., $\varepsilon_t = 1/t$), which can achieve logarithmic regret asymptotically. However, even with decay, the exploration is not directed: the algorithm may still waste pulls on arms that have been proven inferior. Thompson Sampling achieves the optimal O(log T) regret without tuning an exploration schedule, as the posterior variance naturally decays with the number of pulls [7]. Empirical comparisons have shown that Thompson Sampling's regret is often an order of magnitude lower than that of ε-greedy on problems with more than two arms [8].

5.4. Computational trade-offs

In terms of per-step computation, ε-greedy requires O(k) operations to find the maximum estimated value. Thompson Sampling also requires O(k) operations (one sample per arm from the Beta distribution), but the constant factor is larger because sampling from a Beta distribution involves evaluating log-gamma functions or using rejection sampling. For very large k (e.g., thousands of arms) and real-time constraints, this difference can matter. However, for typical recommendation or A/B testing scenarios ($k \leq 100$), the computational overhead of Thompson Sampling is negligible.

Based on the above analysis, the following practical guidelines are suggested. First, ε-greedy is used when computational resources are extremely limited, the number of arms is very large ($k > 1000$) and each pull must be sub-millisecond, or a simple baseline is needed, which is easy to explain to non-technical stakeholders. Second, Thompson Sampling is needed when sample efficiency is important (e.g., each pull has a real cost), the horizon is long ($T >> k$), you can afford to maintain posterior parameters, and you have reasonable prior information or can use uninformative priors.

6. Empirical results and discussion

To evaluate the two algorithms, this study conducted simulations in a Bernoulli multi-armed bandit setting with three arms having true success probabilities $[0.3,0.5,0.7]$. The horizon was $T = 5000$ steps, and $\varepsilon = 0.1$ for ε-greedy. Thompson Sampling used $\mathrm{Beta}(1,1)$ priors. Results are shown in Figure 1.

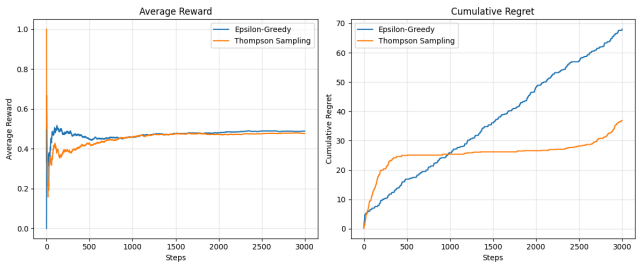

Figure 1. Comparison of ε-greedy and Thompson Sampling. (a) Average reward (b) Cumulative regret

DOI: 10.54254/2755-2721/2026.CH33252

Figure 1(a) shows that Thompson Sampling initially exhibits slightly higher regret due to aggressive early exploration. However, after approximately 1000 steps, its regret growth slows dramatically, while ε-greedy continues to accumulate regret at a nearly linear rate. This demonstrates that Thompson Sampling quickly identifies the optimal arm and reduces exploration, whereas ε-greedy wastes exploration on suboptimal arms indefinitely. Simulation results are consistent with the findings of Le., which also reported superior performance of Thompson Sampling in small-arm settings [8].

Figure 1(b) shows average reward over time. Both algorithms converge to similar levels, but Thompson Sampling achieves a marginally higher and more stable average reward in later stages. The difference in cumulative regret is much more pronounced than the difference in average reward, indicating that regret is a more sensitive metric for learning efficiency.

Overall, these results confirm the theoretical analysis: Thompson Sampling balances exploration and exploitation more effectively, leading to superior long-term performance.

However, this study has several limitations. It primarily relies on theoretical analysis and simulated experiments; real-world validation with large-scale datasets is lacking. Additionally, we only considered Bernoulli rewards and fixed horizons. The comparison of computational overhead and sensitivity to hyperparameters (e.g., ε value, prior choices) was not thoroughly examined. Furthermore, this study did not consider non-stationary environments or delayed feedback scenarios, which are common in real-world applications.

Future research could extend this work by evaluating both algorithms on real-world datasets, such as online click-through rate data, and by exploring non-stationary environments or contextual bandit settings. One promising direction is to evaluate both algorithms in environments with delayed feedback, as done by [9]. Another interesting avenue is to investigate hybrid approaches that combine the simplicity of ε-greedy with adaptive exploration mechanisms. Recent advances such as ε-exploring Thompson Sampling suggest that the boundary between these two classes of algorithms may be productively blurred [10].

7. Conclusion

This paper compares the ε-greedy strategy and Thompson Sampling for multi-armed bandit problems. Thompson Sampling uses Bayesian inference to adaptively explore based on posterior uncertainty, achieving a more effective exploration-exploitation trade-off. The ε-greedy method, while simple and computationally efficient, relies on fixed random exploration, which can lead to unnecessary regret. Empirical simulations confirm that Thompson Sampling accumulates significantly lower cumulative regret over time. A side-by-side comparison table highlights key differences in adaptivity, computational cost, and asymptotic regret.

The multi-armed bandit framework continues to be a vibrant area of research, and understanding the trade-offs between simple and sophisticated algorithms remains crucial for practical deployment. This study provides useful guidance for researchers and practitioners facing sequential decision-making problems under uncertainty.

References

[1] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press, Cambridge, MA.

[2] 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.

DOI: 10.54254/2755-2721/2026.CH33252

[3] Chapelle, O., & Li, L. (2011). An empirical evaluation of Thompson sampling. In Advances in Neural Information Processing Systems (NIPS 2011) (pp. 2249–2257).
[4] Letard, A., Gutowski, N., Camp, O., & Amghar, T. (2024). Bandit algorithms: A comprehensive review and their dynamic selection from a portfolio for multicriteria top-k recommendation. Expert Systems with Applications, 246, Article 123207.
[5] Robbins, H. (1952). Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 58(5), 527–535.
[6] Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on Thompson sampling. Foundations and Trends in Machine Learning, 11(1), 1–96.
[7] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47(2), 235–256.
[8] Le, D. H. (2023). Exploration-exploitation trade-off approaches in multi-armed bandit. Master's thesis, Department of Information Technology, Uppsala University.
[9] Liu, K., & Maghsudi, S. (2024). Budgeted recommendation with delayed feedback. arXiv preprint, arXiv: 2405.11417.
[10] Jin, T., Yang, X., Xiao, X., & Xu, P. (2023). Thompson Sampling with less exploration is fast and optimal. In Proceedings of the 40th International Conference on Machine Learning (ICML 2023) (pp. 15200–15217).