Multi-Armed Bandit Algorithms for Large Language Model Optimization: A Survey of Theory and Applications

Xiangling *Yang*[*]

Aberdeen Institute of Data Science and Artificial Intelligence, South China Normal University (Nanhai Campus), Foshan 528225, Guangdong, China

Abstract. This article systematically expounds on the application of multi-armed bandit (MAB) algorithms in the optimization of large language models (LLMs). Facing the core challenge of the vast search space and high evaluation cost in the prompt engineering, hyperparameter tuning, and generation strategy selection of LLM, MAB has offered a powerful mathematical framework, offering a theoretically guaranteed efficient way to find the optimal solution with the 'Explore then Commit' strategy. The paper details the classic explore-exploit algorithms, including ε -first algorithm (ETC), upper confidence bound algorithm (UCB), asymptotic UCB algorithm and its variants, Thompson sampling (TS), etc., and analyze their specific applications in scenarios such as prompt optimization, hyperparameter tuning, and response generation for LLMs. Through theoretical derivation and case validation, this paper clarifies how to combine the rigor of Bandit theory with the complexity of LLM applications, providing a highly promising technical path for achieving efficient, adaptive, and automated optimization of LLMs.

1 Introduction

One of the most fundamental problems in online decision-making is how to balance the exploration of unknown options and the exploitation of known advantages. The multi-armed bandit model is a classic formal framework to formalize this problem. With the recent large language models (LLMs) being widely used in natural language processing tasks, how to apply bandit algorithms in the training, inference, and optimization of LLMs has become an important method to improve the performance and efficiency of LLMs. In the optimization of LLMs, each "arm" can be treated as a candidate option (e.g., the prompt template, the hyperparameter configuration, and the generation strategy), and the reward is some quality metrics of LLM's output (e.g., accuracy, user feedback, and BLEU score). In this paper, the authors systematically review the classical bandit algorithms (e.g., Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), and Thompson Sampling (TS)) and their application paradigms in LLMs.

[*] Corresponding author: x.yang1.23@abdn.ac.uk

This paper summarizes the cross-disciplinary development process of the research on Bandit theory and LLM optimization. The study explain the theoretical background of the classic Bandit algorithm in detail, including Explore-Then-Commit (ETC) strategy, UCB and its asymptotically optimal version, and Thompson Sampling (TS), and focus on explaining their specific implementation scenario in LLM optimization. Based on the combination of theoretical derivation and case validation, this paper provides systematic methodological guidance and implementation solution to promote the deep connection of Bandit and LLM, and provides important theoretical basis and application route to promote the adaptive optimization of LLM.

The application of Bandit algorithm in LLM field can not only effectively improve the performance of model, but also greatly reduce the consumption of human-computer resources in the optimization process. Especially in the specific scene of prompt engineering, hyperparameter tuning, and generation strategy selection, Bandit uses its intelligent exploration-exploitation balance method to provide a feasible technical solution to automatically optimize LLM. It is hoped that this paper can promote the further development of this cross-discussion field, and provide new ideas and methods to build a more intelligent and efficient LLM optimization system.

2 The Basics and Classic Algorithms of Multi-Armed Bandits

2.1 Formalization of the MAB

Suppose there are K arms in total, each with an unknown reward distribution μi (following by the independent distribution Pa), the average reward for arm a is: $\mu_a = E[X_t | A_t = a]$, $X_t \in [0,1]$; The maximum reward (best arm) is: $\mu^* = \max_{a \in A} \mu_a$. In each round t, the agent selects one arm a_t , and gain the reward. Our objective is to maximize the stimulate reward or to minimize stimulate regrets equivalently:

$$R(T) = \sum_{t=1}^{T} (\mu^* - \mu_{at}) \quad (1)$$

The total regret can be decomposed into the suboptimal contributions of each arm:

$$R_n = \sum_{a \in A} \Delta_a E[T_a(n)] \quad (2)$$

where $\Delta_a = \mu^* - \mu_a$ (the suboptimal gap of arm a), $T_a(n)$ is the number of time to select the arm a in the formal n rounds.

This framework is directly applicable to LLM optimization, for instance, mapping arms to prompt templates or hyperparameter configurations, with rewards corresponding to output quality metrics such as BLEU scores or user satisfaction.

2.2 Explore-Then-Commit (ETC) algorithm

Core idea: Conduct pure exploration (try each arm m times) at first, and then select the arm with the highest empirical average reward for exploitation.

Exploration Phase: In the first $m * k$ rounds (where k is the number of arms), each arm is fixedly selected m times to collect reward data and estimate the average reward. With the exploration regret (loss of rewards due to choosing the suboptimal arm):

$$R_{explore} = \sum_{a=}^{k} m\Delta_a \quad (3)$$

Exploitation Phase: Starting from the $(m * k + 1)$ round, the arm with the highest average reward estimated in the exploration phase is always selected to maximize the cumulative reward. With the exploitation regret (misjudged the optimal arm, thereby lead to continuous losses):

$$R_{exploit} = (n - m)\Delta_{\hat{a}} P_{error} \quad (4)$$

where â is the chosen suboptimal arm, P_{error} is the possibility of misjudgment.

Algorithm 1: Explore-Then-Commit (ETC) Pseudocode

Here are the detailed pseudo-code:

Input: Number of arms K, total rounds T, exploration count per arm m
Initialize: For each arm $a \in [K]$, set $Sa \leftarrow 0$, $Na \leftarrow 0$
for t = 1 to m × K do //exploration phase
$A_t \leftarrow t \bmod K$ //arm selection
Observe reward $X_t \sim P_{at}$
$S_{at} \leftarrow S_{at} + X_t$
$N_{at} \leftarrow N_{at} + 1$
end for
for each arm a ∈ [K] do
$\hat{\mu_a} \leftarrow S_a / N_a$
end for
$a^* \leftarrow \text{argmax}_a \hat{\mu_a}$
for t = mK + 1 to T do //exploitation phase
$A_t \leftarrow a^*$
Observe reward $X_t \sim P_{at}$
end for

2.3 Upper Confidence Bound(UCB) algorithm

Core idea (Optimism in the Face of Uncertainty): In each round, for each arm an "Upper Confidence Bound" was calculated, which is the current empirical average reward plus a reward bias based on uncertainty (exploration reward). The algorithm always selects the arm with the highest upper bound. This mechanism ensures that after an arm is fully explored, its uncertainty decreases and the exploration reward reduces; while arms that have not been fully explored, due to high uncertainty, have a large exploration reward and thus have the opportunity to be selected.

Construction of confidence intervals: For the arm with 1-degree Gaussian distribution, the confidence interval of its empirical average reward $\hat{\mu}_{i,t}$ after t rounds can be derived from the Hoeffding inequality:

$$P\left(\mu_i \geq \hat{\mu}_{i,t} + \sqrt{\frac{2\log\left(\frac{1}{\delta}\right)}{T_i(t)}}\right) \leq \delta \tag{5}$$

where $T_i(t)$ is the times arm i being chosen until t round. UCB index value: In the t round, each arm i computes an index value:

$$UCB_{i,t-1} = \hat{\mu}_{i,t-1} = \sqrt{\frac{2\log\left(\frac{1}{\delta}\right)}{T_i(t-1)}}) \tag{6}$$

By setting δ=1/t (the confidence level that decays over time), a practical UCB1 index is obtained:

$$UCB_{i,t-1} = \hat{\mu}_{i,t-1} = \sqrt{\frac{2\log t}{T_i(t-1)}}) \tag{7}$$

The algorithm choosing the arm with maximal index value: $A_t = \arg\max_i UCB_{i,t-1}$[2].

Algorithm 2: Upper Confidence Bound (UCB) Pseudocode

Here is the detailed pseudo-code:

Input: K, T, exploration parameter $c > 0$
Initialize: $\forall a \in [K]$, $S_a \leftarrow 0$, $N_a \leftarrow 0$
for t = 1 to K do //Initial Phase
$A_t \leftarrow t$
Observe X_t
$S_{at} \leftarrow S_{at} + X_t$
$N_{at} \leftarrow N_{at} + 1$
end for
for t = K+1 to T do //Main loop
for each arm $a \in [K]$ do //Loop over each arm to calculate UCB index
$\hat{\mu_a} \leftarrow S_a / N_a$
$UCB_a \leftarrow \hat{\mu_a} + c \times \sqrt{\left(\frac{\log t}{N_a}\right)}$
end for
$A_t \leftarrow \text{argmax}_a\ UCB_a$
Observe X_t
$S_{at} \leftarrow S_{at} + X_t$ //Update cumulative reward for selected arm
$N_{at} \leftarrow N_{at} + 1$ //Update pull count
end for

2.4 Asymptotically Optimal UCB algorithm

Core idea: Asymptotically optimal UCB variants (e.g., KL-UCB) are designed to achieve the theoretical lower bound established by Lai and Robbins (1985) [1], characterized by a regret growth rate of $O(\log T)$. Unlike standard UCB, which employs confidence intervals derived from Hoeffding's inequality, these asymptotically optimal variants utilize tighter probabilistic bounds, such as the Kullback-Leibler (KL) divergence, to more precisely balance exploration and exploitation [2, 3].
KL-UCB index value:

$$KL - UCB_{i,t} - \max\{q \in [0,1]: KL(\hat{\mu}_{i,t}||q) \leq \frac{\log t + c\log\log t}{T_i(t)}\} \quad (8)$$

where $KL(\hat{\mu}_{I,t}||q)$ is the KL divergence between Bernoulli distribution: $KL(\hat{\mu}_{i,t}||q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ parameter c>0 is a constant [4].

2.5 Thompson Sampling (TS) algorithm

Core idea: Based on Bayesian theory, it resolves the exploration-exploitation trade-off through random sampling and posterior update. Unlike the frequentist UCB algorithm, TS adopts a Bayesian approach, treating the reward distribution of each arm as a random variable and guiding decisions by maintaining its posterior distribution. The core idea of TS is to sample a reward estimate from the posterior distribution of each arm and then select the arm with the highest sampled value. This method naturally balances exploration (arms with potentially higher sampled values) and exploitation (arms with higher true means) [5]. Learning framework: Prior Distribution: Initialize a prior (e.g., mean and variance of a Gaussian distribution) for the reward distribution of each arm. Posterior Update: After selecting an arm and observing the reward in each round, update the posterior distribution

according to Bayes' theorem. Sampling Decision: Sample from the updated posterior distribution and select the arm with the highest sampled value. For a Gaussian likelihood with known variance σ^2 and a Gaussian prior $\mathcal{N}(\mu_0, \sigma_0^2)$, the posterior distribution after $T(t)$ observations with mean $\hat{\mu}$ is Gaussian: $\mathcal{N}\left(\mu_i(t), \sigma_i^2(t)\right)$. The TS algorithm possesses asymptotic optimality. For Guassian muti-armed bandits, the regret bound of TS is:

$$\lim_{t\to\infty} \sup \frac{R_t}{\log t} \leq \sum_{i:\Delta i>0} \frac{\Delta i}{KL(\mu_i, \mu^*)} \tag{9}$$

where $\Delta i = \mu^* - \mu_i$ is the gap between optimal and suboptimal arm [6].

Algorithm 3: Thompson Sampling (TS) Pseudocode

Here is the detailed pseudo-code:

```
Input: K, T
Initialize: ∀a ∈ [K], α_a ← 1, β_a ← 1
for t = 1 to K do   //Main loop: iterate over all time steps
for each arm a ∈ [K] do
Sample θ_a ~ Beta (α_a, β_a)
end for
A_t ← argmax_a θ_a   //Select the arm with the highest sampled value
Observe reward X_t ∈ {0, 1}
if X_t == 1 then   //if reward is success (1)
α_at ← α_at + 1
else   //if reward is failure (0)
β_at ← β_at + 1
end if
end for
```

3 The combined application of Bandit and LLM

The integration of Multi-Armed Bandit (MAB) algorithms with Large Language Models (LLMs) offers an efficient and adaptive solution for optimizing multiple critical aspects of LLMs [7]. This section provides a detailed examination of the specific applications of Bandit algorithms in prompt optimization and hyperparameter tuning [8].

3.1 Bandit Algorithms for Prompt Optimization

Prompt optimization serves as a pivotal technique for eliciting the capabilities of Large Language Models (LLMs), with the objective of enhancing output quality through the design of effective prompt templates. Bandit algorithms automate this process via intelligent exploration and exploitation mechanisms, substantially reducing the need for manual trial-and-error efforts. Application scenarios and problem formalization [9]. In the optimization of LLM prompts, each candidate prompt template can be regarded as an "arm". The reward function is designed based on the quality metrics of LLM's output, such as:

- For text generation tasks: automatic evaluation metrics like BLEU and ROUGE-L are adopted.
- For classification or question-answering tasks: accuracy, F1 score, etc. are used.

- In human feedback scenarios: user satisfaction scores or manual assessment scores are utilized.

The rewards need to be normalized to the [0,1] interval to fit the standard framework of the Bandit algorithm. Suppose that the reward of the i-th prompt template is x_i , the expected reward is $\mu_i = E[x_i]$, the aim is to minimize the accumulated regret $R_n = n\mu^* - \sum_{t-1}^{n} E[x_i]$, where $\mu^* = max_i \mu_i$.

Efficiency Enhancement: Bandit algorithms rapidly converge to high-performance prompts while minimizing evaluation overhead. The theoretical regret bound is $O(\log n)$, substantially outperforming the $O(n)$ bound of random search.

Adaptive Optimization: In dynamic environments (e.g., shifting user preferences), algorithms such as Thompson Sampling (TS) dynamically adjust strategies via posterior updates, ensuring sustained high performance.

Scalability: These methods seamlessly integrate into complex scenarios including multi-prompt compositions and Chain-of-Thought (CoT) prompting, enabling comprehensive optimization of both prompt structures and parameters.

3.2 Algorithm Selection

Explore-Then-Commit (ETC): Simple and easy to implement, suitable for scenarios with stable reward distribution and a small number of arms; the theoretical regret bound is $O(K\Delta logT)$. The applicable scenarios of LLM: When the parameter space is small and the training cost is high in hyperparameter tuning, it can quickly screen candidate configurations.

Upper Confidence Bound (UCB): No need for prior distribution, tight regret bound $\left(O\left(\sqrt{KTlogT}\right)\right)$; Adaptable and capable of balancing exploration and exploitation. LLM application scenarios: When the reward is stable in prompt optimization (such as based on automatic metrics like BLEU) and real-time decision-making is required.

Thompson Sampling (TS): Asymptotically optimal, with a regret bound of $O(logT)$; naturally handles uncertainty and is robust in non-stationary environments. LLM application scenarios: when the reward distribution is complex or dynamically changing (such as user feedback data), or when it is necessary to quickly adapt to distribution drift.[7][8]

4 Future challenges

4.1 Adaptability in Non-stationary Environments

The reward distribution of LLMs may change over time (such as data distribution drift or user preference changes), requiring Bandit algorithms to have rapid adaptability. Existing methods such as sliding window UCB or discounted TS can partially alleviate this issue, but theoretical guarantees and experimental verification are needed. In the future, more powerful adaptive algorithms, such as change-detection-based Bandit or reinforcement learning ensembles, need to be developed.

4.2 Trade-offs in Multi-objective Rewards

The output quality of LLMs needs to be evaluated from multiple dimensions, including accuracy, fluency, safety, fairness, etc. There are often trade-offs among these objectives, and a single reward function is difficult to capture this complexity. Designing a Bandit framework that can simultaneously optimize multiple competing objectives faces dual

challenges in theory and methods, requiring balancing the importance of different objectives and handling possible conflicts.

5 Conclusion

The Bandit algorithm offers a promising solution for the optimization of large language models (LLMs), with its core value lying in the ability to significantly enhance the efficiency and effectiveness of LLMs in key areas such as prompt optimization and parameter tuning through an intelligent exploration-exploitation trade-off mechanism. Theoretical analysis indicates that algorithms like UCB and Thompson Sampling can achieve good regret bounds under appropriate conditions, providing a theoretical foundation for practical applications.

However, it must be acutely aware that the successful application of Bandit algorithms in the LLM domain still has a long way to go. The current challenges, such as adapting to non-stationary environments, extending to high-dimensional spaces, and addressing multi-objective optimization, all require in-depth theoretical innovations and systematic methodological breakthroughs. The complexity and interrelatedness of these challenges demand a cross-disciplinary research approach that integrates knowledge from multiple fields, including machine learning, optimization theory, and systems engineering.

The integration of Bandit algorithms with LLMs represents a significant direction in the development of automated machine learning. Through continuous theoretical innovation and methodological breakthroughs, Bandit algorithms are expected to play a more crucial role in the LLM ecosystem, not only promoting the advancement of LLM technology itself but also providing key technical support for building more intelligent, efficient, and reliable artificial intelligence systems. The in-depth development of this research direction requires close collaboration between academia and industry to jointly address theoretical challenges and engineering practice issues.

References

1. T. L. Lai, H. Robbins, Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.***6**, 4–22 (1985).
2. P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.***47**, 235–256 (2002).
3. W. R. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika***25**, 285–294 (1933).
4. O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, G. Stoltz, Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Ann. Stat.***41**, 1516–1541 (2013).
5. S. Agrawal, N. Goyal, Analysis of Thompson Sampling for the multi-armed bandit problem. *Conf. Learn. Theory*39.1–39.26 (2012).
6. D. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen, A tutorial on Thompson sampling. *Found. Trends Mach. Learn.***11**, 1–96 (2018)
7. L. Li, W. Chu, J. Langford, R. E. Schapire, A contextual-bandit approach to personalized news article recommendation. *Proc. Int. Conf. World Wide Web*661–670 (2010).
8. J. Snoek, H. Larochelle, R. P. Adams, Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.***25** (2012).
9. Y. Deng, et al., An exploration of bandit algorithms for automatic prompt optimization in large language models. *Proc. Conf. Empir. Methods Nat. Lang. Process.*(2023).