Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/13/20230705

Empirical performances comparison for ETC algorithm

Tianfeng Chen

School of Computer Science, University of Nottingham, United Kingdom, NG8 1BB

scytc1@nottingham.ac.uk

Abstract. Explore-then-commit (ETC) algorithm is a widely used algorithm in bandit problems, which are used to identify the optimal choice among a series of choices that yield random outcomes. The ETC algorithm is adapted from A/B testing, a popular procedure in decision-making process. This paper explores the multi-armed bandit problem and some related algorithms to tackle the multi-armed bandit problem. In particular, this paper focuses on the explore-then-commit (ETC) algorithm, a simple algorithm that has an exploration phase, and then commits the best action. To evaluate the performance of ETC, a variety of settings is made in the experiment, such as the number of arms and input parameter m, i.e., how many times each arm is pulled in the exploration phase. The result shows that the average cumulative regret increases when the number of arms gets larger. With the increase of parameter m, the cumulative regret decreases in the beginning, until reaching the minimum value, and then starts increasing. The purpose of this paper is to empirically evaluate the performance of the ETC algorithm and investigate the relationships between the parameter settings and the overall performance of the algorithm.

Keywords: multi-armed bandit, stochastic bernoulli bandit, explore-then-commit algorithm.

1. Introduction

The multi-armed bandit problem is one of the most popular and important problems in machine learning and probability theory, which exemplifies the classical exploration-exploitation dilemma. A bandit problem is a sequential game between a learner and an environment [1]. There are a wide range of applications for bandit algorithms, for example, recommendation systems, clinical trials and the selection of optimal machine learning model to deploy in technology companies. A variety of bandit models exist as well, such as stochastic bandits, adversarial bandits, contextual bandits and combinatorial bandits, depending on the type of problem to be solved [1]. In this paper, stochastic bandits will be discussed in detail, which is a sequential decision-making process where the reward of each arm follows a certain probability distribution.

There exists a spectrum of algorithms that can be applied to tackle the bandit problem, one of which is called explore-then-commit (ETC), a simple but effective algorithm to solve stochastic bandit problems. The idea of ETC algorithm is originated from A/B testing, a popular user experience research methodology. It is vital to understand the performance of each algorithm, therefore a spectrum of papers have already discussed the mathematical proofs of the ETC algorithm performance. However, the empirical study of the explore-then-commit algorithm is scarce, in particular, the relationship between the input parameter m and the performance outcome. Therefore, this paper

DOI: 10.54254/2755-2721/13/20230705

mainly evaluates the performance of ETC algorithm from an empirical research perspective. It begins by recapitulating the rigid definition of bandit problem and the ETC algorithm. After that, it describes the example environment as well as the setup of experiment and parameter. Finally, it demonstrates the result of the experiment and provides some explanations of the result afterwards.

2. Literature review

A number of mathematical proofs and regret analyses on the explore-then-commit algorithm can be found in the bandit algorithm textbooks [1, 2], which provide a thorough, theoretical examination of the explore-then-commit algorithm's performance. In addition, Garivier et al. [3] suggest a method to achieve asymptotically optimal performance of the ETC algorithm without previous knowledge of sub-optimality gaps between each arm. Jin et al. [4] propose a double explore-then-commit (DETC) algorithm that has two exploration phases and two exploitation phases, which achieves asymptotic optimality. They also provide theoretical proofs and empirical experiments of the algorithm. As for Kuleshov and Precup's research [5], they conducted a series of empirical confirmations of the effectiveness of classical multi-armed bandit algorithms. Since the ETC algorithm is largely inspired by A/B testing, a variety of research on A/B testing performance exists as well. Gui et al. [6] and Kaufmann et al. [7] examine the complexity of A/B testing algorithm from a mathematical perspective. Xu et al. [8] focus on A/B testing at a large scale in social network settings, such as LinkedIn. Young [9] applies A/B testing to a practical web-based application in an academic library, while Gilotte et al. [10] evaluate the offline A/B testing performance and propose a new counterfactual estimator.

3. Methodology

A bandit problem is a sequential game between a learner and an environment [1]. An environment consists of a set of actions. Each action reveals a reward when it is pulled. The game will be played in n rounds in total (also known as the **horizon**). In each round $t = 1, 2, \ldots, n$, the learner chooses a bandit arm (action) A_t from a set of k possible actions. When the arm is pulled by the learner, the environment will reveal a reward X_t, according to the probability distribution of the chosen arm. The objective of the learner is to maximize the cumulative reward over n rounds, i.e., $\Sigma_{t=1}^{n}$. The main challenge for the learner is that it is impossible to obtain the future reward of each arm in advance. The only knowledge for the learner is the list of action-reward history ($A_1, X_1, \ldots, A_{t-1}, X_{t-1}$). To maximize the cumulative reward, the learner has to choose the optimal action from the action list according to the history only.

A common and simple idea for the learner to obtain the optimal action at round t would be to select the arm which has the highest average reward from previous t-1 rounds, since the arm with the highest average reward in history is more inclined to produce a higher reward in the following round, by intuition. This is also known as exploitation, i.e., taking advantage of the previous knowledge to choose the favorable action in the next round. However, since the expected rewards for the arms are hidden from the learner, there can be severe variance between the average reward obtained from the first several rounds and the actual mean reward. Without enough exploration, the learner is likely to be trapped in the local optimum, and therefore miss the best optimal action. Hence, the perfect balance of exploration and exploitation is, by necessity, vital in the performance improvement of multi-armed bandit algorithms.

To quantify the performance of bandit algorithms, some performance measures are required to be defined appropriately. One of the most famous measurement criteria is **regret**. Assuming that the action set $\mathcal{A} = 1,2,\ldots,k$, this study defines μ_a to be the mean reward of arm a, and the optimal action $a^* = argmax_{a\in\mathcal{A}}\mu_a$ is the action that produces the maximum mean reward, and the maximum mean reward is denoted by $\mu^* = max_{a\in\mathcal{A}}\mu_a$. The reward obtained at round t is denoted by X_t. Therefore, the regret over n rounds becomes $R_n = n \max_{a\in\mathcal{A}} - \mathbb{E}[\sum_{t=1}^{n} X_t]$, where the first term refers to the maximum mean reward the learner can obtain in the first n rounds, and the second term refers to the expected reward obtained by the learner in the experiment during the first n rounds [1].

DOI: 10.54254/2755-2721/13/20230705

Before the experiment, a brief overview of the bandit algorithm is demonstrated below. The first algorithm is called the **explore-then-commit** (ETC) algorithm, which is inspired by the idea from A/B testing, i.e., pulling each action a fixed number of times in a round-robin way during the exploration phase, and then committing the action with the maximum average reward in the commit phase. The definition of $T_i(t)$, $\hat{\mu}_i$, as well as the rigid definition of ETC algorithm are given below.

Algorithm 1 **Explore-then-commit [1]**

1: Input m.

2: In round t choose action

$$A_t = \begin{cases} (t \bmod k) + 1, & t \leq mk; \\ argmax_i \hat{\mu}_i(mk), & t > mk. \end{cases}$$

There are large varieties of bandit algorithms, which can be found in bandit algorithm textbooks [1, 2]. However, this paper focuses on the performance of the ETC algorithm. The other algorithms are provided here only for reference and performance comparison.

3.1. Example environment

The prominent characteristics of the bandit problem are the number of arms and the reward distribution of each arm, and it turns out that these two characteristics also become the only characteristics of a bandit problem that need to be considered [5]. For simplicity, a two-armed **stochastic Bernoulli bandit** is be experimented. A stochastic Bernoulli bandit is a subset of a stochastic bandit in which the reward $X_t = \{0,1\}$ and there exists a vector $\mu \in [0,1]^k$ such that the probability of $X_t = 1$ given the action selected at round t $(A_t = a)$ is μ_a [1]. From the above definition, it can be found that the outcome of the reward for each arm is either 0 or 1, and the probability of arm a producing 1 as the reward is denoted by μ_a.

Stochastic Bernoulli bandits are ubiquitous around the world. For example, assume that there is an advertisement company that created two different web pages. The objective is to figure out which web page is more appealing to the user, such that the user is more likely to click the web page and look through the content. One of the two web pages will be distributed to the user according to a certain policy, and the user can choose whether to click that web page or not. If the user chooses to click that web page, it will reveal a reward of 1. Otherwise, the reward will be 0. In this scenario, the index of the web page delivered to the user is an action, and the choice of the user is the environment. The final objective is to develop a web page distribution algorithm that maximizes the cumulative reward. It is also assumed that the choice of the user is a Bernoulli distribution. For instance, assume that the expected reward of arm 1: $\mu_1 = 0.3$, it suggests that only 30% of the users who have seen the web page choose to click it, while the remaining 70% choose not to click it. This is a typical bandit problem, which can be experimented by classical bandit algorithms.

3.2. Experimental setup

In the following experiments, two types of Bernoulli bandits will be examined: two-armed Bernoulli bandits and five-armed Bernoulli bandits. Note that by Hoeffding's lemma, Bernoulli distribution is ½-subgaussian, and therefore 1-subgaussian.

As for the two-armed Bernoulli bandits, this paper defines the success probability of each arm $p = [0.3, 0.7]$, or equivalently, the expected reward for each arm $\mu = [0.3, 0.7]$. It is straightforward to find that the optimal arm $a^* = argmax_{a \in \mathcal{A}} \mu_a = 3$, and the maximum expected reward $\mu^* = max_{a \in \mathcal{A}} \mu_a$. This study also defines the sub-optimality gaps for each arm, which are calculated by the difference between the maximum expected reward and the expected reward of that arm, i.e., $\Delta_a = \mu^* - \mu_a$. For the optimal arm, the sub-optimality gap $\Delta_{a^*} = 0$, and the sub-optimality gaps for each arm $\Delta_a = [0.4, 0]$.

As for the five-armed Bernoulli bandit, this study defines the success probability of each arm $p = [0.1, 0.3, 0.5, 0.7, 0.9]$, and the expected reward for each arm $\mu = [0.1, 0.3, 0.5, 0.7, 0.9]$. Similarly, it

DOI: 10.54254/2755-2721/13/20230705

can be deduced that for the five-armed bandit, $a^* = argmax_{a \in \mathcal{A}} \mu_a = 5$, and $\mu^* = max_{a \in \mathcal{A}} \mu_a = 0.9$. The sub-optimality gaps for each arm $\Delta_a = [0.8, 0.6, 0.4, 0.2, 0]$.

3.3. Parameter setting

For simplicity, the total number of rounds i.e., the horizon in the following experiment, is set to be 100,000. There exists only one parameter in ETC, that is *m*, the number of times each arm is pulled during the exploration phase. With regards to the two-armed bandit, if the sub-optimality gaps are already known to the learner, the optimal *m* can be calculated by the formula $\left\lceil \frac{4 \log n}{\Delta^2} \right\rceil$. Therefore, we try to evaluate the algorithm on settings where *m* equals 50, 100, 200, 500, and optimal. In terms of the five-armed bandit, since the optimal *m* formula only applies to the two-armed bandit, the parameter *m* is set to be 50, 100, 200, 500, and 1000 correspondingly.

Note that the training data for this experiment will be sampled from Bernoulli distribution using the artificial pseudo-random number, rather than real-world data set.

3.4. Evaluation criteria

The following performance criteria will be evaluated in this experiment:

Cumulative regret over the experiment (R_n)

The percentage of plays in which the optimal arm is committed

Each experiment was repeated 1000 times, and the results were the average value of each round.

3.5. Results

The main results are shown in Figure 1, Figure 2 for the two-armed bandit, and Figure 3, Figure 4 for the five-armed bandit. The *x*-axis represents the current round number, while the *y*-axis represents the corresponding average cumulative reward at each round.

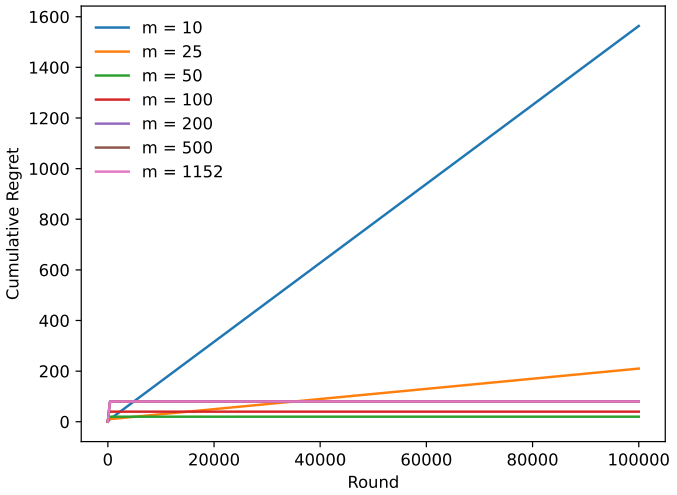

Figure 1. R_n of 2-armed bandit.

DOI: 10.54254/2755-2721/13/20230705

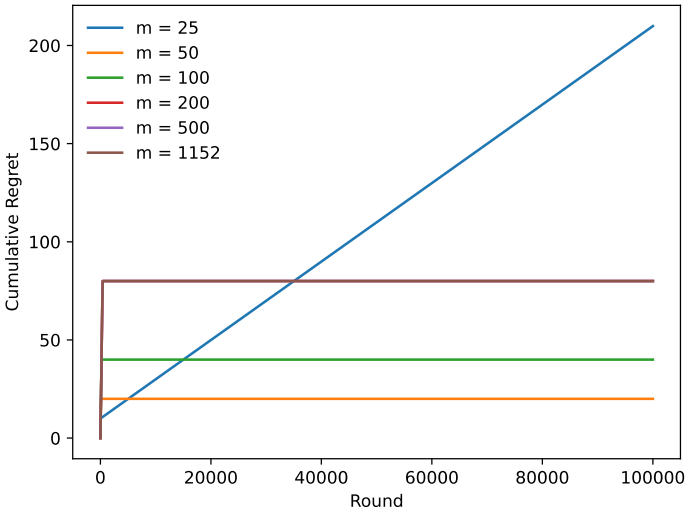

Figure 2. R_n of 2-armed bandit (except m = 10).

With regards to two-armed bandit, it can be found that from m = 10 to 50, the growth rate of cumulative regret decreases rapidly, while from m = 50 to optimal (1152), the growth rate is approximately the same, with some steady increase. Note that the linear growth of cumulative regret curves for m = 10 and 25 is evident, while for m greater than 25, the curve sharply grows at first hundreds of rounds and then stabilize without significant increase.

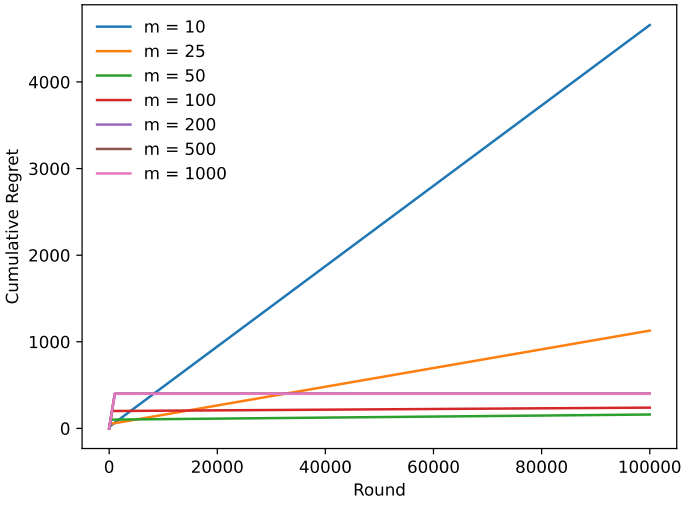

Figure 3. R_n of 5-armed bandit.

DOI: 10.54254/2755-2721/13/20230705

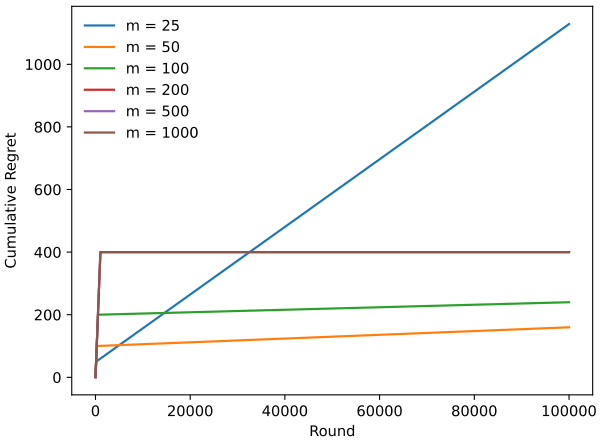

Figure 4. R_n of 5-armed bandit (except m = 10).

Compared to two-armed bandit, the average cumulative regret of five-armed bandit suggested in the graph for each m is relatively higher. However, similar to two-armed bandit, the growth rate decreases from m = 10 to 50, and then increase steadily from 50 to 1000. In addition, except for m = 10 and 25, the slope of the remaining curves is approximately equal to 0.

Table 1. Percentage of committing the optimal arm (Unit: %).

k \ m	10	25	50	100	200	500	1000	1152
2	96.1	99.5	100	100	100	100	-	100
5	78.3	94.7	99.7	99.8	100	100	100	-

Table 1 evaluates the percentage of plays in which the optimal arm is committed, in relation to the number of arm k and input parameter m. The table indicates that, overall, with regards to the percentage of committing the optimal arm, five-armed bandit is generally lower than two-armed bandit. It is also suggested that when m = 10, two-armed bandit has already achieved more than 95% accuracy; while for five-armed bandit, m should at least be greater than 25.

4. Discussion

For many bandit algorithms, such as UCB and asymptotically optimal UCB, they can be proved to achieve logarithmic regret, i.e., $R_n = O(\log n)$. The cumulative regret curves of these algorithms clearly display the logarithmic growth shape. However, in terms of the shape of the curve, this is not the case for the ETC algorithms in our experiment, due to the intrinsic characteristics of the ETC algorithm. As mentioned in the introduction section, there is a sharp transition from the exploration phase to the commit phase. In the exploration phase, since each arm is selected equally in turn, the cumulative regret grows linearly. The reason is that each sub-optimal choice will contribute a constant penalty to the regret, while the optimal choice will not contribute any penalty to the regret. Therefore, in each round, the average regret per round will be the average of sub-optimality gaps. This can be exemplified by the steep linear line at the very beginning of the rounds.

Nevertheless, when the exploration phase reaches its end, there is an abrupt change in the slope of the curve. The curve becomes flat immediately. In the five-armed bandit, it is worth noticing that when m is approximately less than 200, the linear growth is clearly evident in the graph, and the slope of the

DOI: 10.54254/2755-2721/13/20230705

curve decreases dramatically when *m* becomes larger. In contrast, if *m* is larger than 500, there is almost no increase in the cumulative regret. In the two-armed bandit, except for relatively smaller *m*, there is almost no increase in cumulative regret for *m* larger than or equal to 50. The result implies that the ETC algorithm tends to behave better when the number of arms is relatively small, which could also be exemplified in the table above. When *m* equals 10, the difference between percentages of optimal arm commitment is significant. However, when *m* is larger or equal to 50, the difference can be negligible. The choices of m should also be considered carefully. If *m* is too small, the probability of committing the sub-optimal choice in the commit phase will become large, which results in a substantial penalty to the cumulative regret. By contrast, if *m* is too large, it will contribute to too much regret in the exploration phase.

There are still many aspects that can be improved in this experiment to obtain more convincing results. For example, because of the variance of pseudo-random numbers, real-world datasets can be applied to the experiment. In addition, since the cumulative regret is calculated by the average of multiple experiments, it is admissible to increase the number of repetitions to gain more precise results. The horizon can also be lengthened to investigate the growth of the curve in larger rounds.

5. Conclusion

This paper briefly discusses the multi-armed bandit problem, a sequential decision-making problem that has a wide range of applications. There exists a variety of classical algorithms to tackle the problem, one of which is the explore-then-commit (ETC) algorithm, which is also relatively straightforward and easy to implement. However, it seems that the empirical analysis of the ETC algorithm is scarce, therefore a set of experiments regarding the ETC algorithm is conducted.

To evaluate the performance of the ETC algorithm, the cumulative regret over the experiment and the percentage of plays in which the optimal arm is committed are examined. The result suggests that the overall performance of the ETC algorithm is significantly affected by the number of arms and the input parameter *m*, and the number of times each action is explored during the exploration phase. There is also a sharp transition from the exploration phase to the exploitation phase. In addition, the linear growth from the graph implies that the ETC algorithm might not be an ideal solution for situations where performance requirement is critical. Overall, this paper provided a thorough empirical analysis of the explore-then-commit algorithm and illustrates the strong relationship between parameter settings and the performance of the ETC algorithm.

References

[1] Lattimore, T. and Szepesvári, C.: Bandit algorithms. Cambridge, United Kingdom: Cambridge University Press, (2020).
[2] Slivkins, A.: Introduction to multi-armed bandits. Boston: Now, (2019).
[3] Garivier, A., Kaufmann, E., Lattimore, T., On Explore-Then-Commit Strategies. Advances in Neural Information Processing Systems 29, (2016).
[4] T. Jin, P. Xu, X. Xiao, and Q. Gu, "Double explore-then-commit: Asymptotic optimality and beyond," in Conference on Learning Theory, PMLR, July, pp. 2584-2633 (2021).
[5] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," arXiv preprint arXiv:1402.6028, 2014.
[6] H. Gui, Y. Xu, A. Bhasin, and J. Han, "Network a/b testing: From sampling to estimation," in Proceedings of the 24th International Conference on World Wide Web, pp. 399–409, 2015.
[7] E. Kaufmann, O. Cappe´, and A. Garivier, "On the complexity of a/b testing," in Conference on Learning Theory, pp. 461–481, PMLR, 2014.
[8] Y. Xu, N. Chen, A. Fernandez, O. Sinno, and A. Bhasin, "From infrastructure to culture: A/b testing challenges in large scale social networks," in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2227–2236, 2015.
[9] S. W. Young, "Improving library user experience with a/b testing: Principles and process," Weave: Journal of Library User Experience, vol. 1, no. 1, 2014.

DOI: 10.54254/2755-2721/13/20230705

[10] A. Gilotte, C. Calauzenes, T. Nedelec, A. Abraham, and S. Dolle, "Offline a/b testing for recommender systems," in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 198–206, 2018.