Empirical evaluation on discounted Thompson sampling for multi-armed bandit problem with piecewise-stationary Bernoulli arms

F C Asyuraa, S Abdullah* and T E Sutanto

1Department of Mathematics, Universitas Indonesia, Kampus Baru UI, Depok 16424, Indonesia
2Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Syarif Hidayatullah, Tangerang Selatan 15412, Indonesia

*sarini@sci.ui.ac.id

Abstract. The Multi-Armed Bandit problem is a problem in reinforcement learning that focuses on how to solve the exploration-exploitation dilemma. The dilemma is, given a set of options (known as arms) that you could try many times, how to balance between gathering information through experimenting with the available arms (exploration) or maximizing profit by choosing the seemingly best arm at that time (exploitation). The Multi-Armed Bandit problem is centered around determining which arm to choose at every round. Multi-Armed Bandit has gained its popularity as a more dynamic approach to a randomized trial, with its goal is to experiment with each available arm while still maximizing profit gained. An example of Multi-Armed Bandit in real life is in determining which film artwork should be shown to a visitor that would attract the visitor to watch that particular film. Bernoulli distribution with parameter $\theta$ is chosen to model the response of the visitor after seeing the artwork. Non-stationary condition on $\theta$ can be implemented to accommodate various trends in film artworks. Some artworks might be good at a certain month, but they could not be preferred in the next month. The non-stationary condition in this study is modeled through piecewise-stationary. We implemented a discounted Thompson sampling policy that used Bayesian method to determine which arm to choose at each round. Multiple simulations were conducted on various conditions to empirically test the policy’s performance on various conditions. Evaluation was based on the cumulative regret. Based on these simulations, discounted Thompson sampling policy achieved relatively lower cumulative regret in tackling the stationary and piecewise-stationary conditions, compared to some well-known policies such as Epsilon Greedy, SoftMax, Upper Confidence Bound, and Thompson Sampling.

1. Introduction

Multi-Armed Bandit (MAB) has gained its popularity as a more dynamic approach to a randomized trial in experimenting over several variants. In MAB problem, rather than allocating participant equally to each variant and choosing the winner (the variant that gives the most reward) at the end of the experiment, the MAB viewed the problem as a sequential one. This means MAB will assign participant sequentially and adapt the allocation as the experiment goes on if some variant shows a good result, as shown in Figure 1.
Figure 1. Comparison of allocation in A/B Testing (left) and Multi-Armed Bandit (right) as the experiment goes on. (Source: Modified from <https://www.dynamicyield.com/lesson/contextual-bandit-optimization/>.)

The dynamic allocation on MAB as shown in Figure 1 made the overall reward achieved higher because participants will be gradually shown less inferior options. To achieve the dynamic allocation of participants in MAB, the participants (also known as rounds in MAB) are experimented sequentially, whereas each participant will be shown one of the variants (also known as arms in MAB) and the response (also known as rewards in MAB) is recorded. This recorded response(s) will be used as a basis for determining which variant to be shown for the next participants as shown in Figure 2. The process of acquiring information sequentially and updating future decision making based on the recorded response made MAB as a reinforcement learning problem [1].

Figure 2. Multi-Armed Bandit example. (Source: modified from J. Basilico, “Artwork Personalization at Netflix”, QCon SF, 2018.)

The goal of MAB problem is to maximize rewards gained in the experimentation, but to achieve that, the learner is faced with a dilemma at each round. Should the learner gather information through experimenting with the available arms (exploration) or should he maximize profit by choosing the seemingly best arm at that time (exploitation)? this is called the exploration-exploitation dilemma [1]. The arm chosen at each round is determined according to a policy that has been adopted by the learner. In MAB problem, numerous policies have been devised to find the balance between exploration and exploitation, some that are well-known are Epsilon Greedy [2], SoftMax [3], Upper Confidence Bound [4], and Thompson Sampling [5].

Multi-Armed Bandit has its applications on various subjects, such as medical trials, web design, advertisement, and recommender systems [6]. Depending on the problem at hand, the reward of each arm could take on many values, it could be continuous, discrete, or even binary [7]. Such examples are shown in Figure 2, the problem at hand is to determine which artwork to show to each visitor, the goal is to show the visitor the most attractive artwork so there is a higher chance that the film will be watched
In this problem, the response of the participants is binary (1 if they watched it, else 0). To simulate this problem, the reward could be generated from Bernoulli distribution with parameter $\theta$ [7].

In the previously mentioned problem, some artworks may receive good attention on a certain month or period of the year, but then lacks the attraction for the rest of the year, or probably some artworks are popular because of a certain holiday or event. This condition is called non-stationary condition [9]. This non-stationarity could be modelled by changing the $\theta$ value on Bernoulli distribution as the times goes on. One of the simpler conditions is the piecewise-stationary condition, that is $\theta$ could change value but remains constant over defined period of time [10]. This piecewise-stationary Bernoulli arms are illustrated in Figure 3.

Figure 3. Piecewise-stationary Bernoulli arms illustration.

In general, the non-stationary condition will be harder to solve because the information gained in the earlier phase of the experiment may not be relevant anymore. Some well-known policies such as Epsilon Greedy [2], SoftMax [3], Upper Confidence Bound [4], and Thompson Sampling [5] are not well suited for the non-stationary condition [11] because those policies used all historical rewards from past actions. Meanwhile, in the non-stationary condition, historical rewards that are long past maybe become irrelevant. Therefore, another policy was needed to tackle non-stationary condition. In this paper, we used Discounted Thompson Sampling policy [12], a discounted version of the Thompson Sampling policy that has been known to be excellent in the stationary Bernoulli arms environment [13].

In its implementation in real-life, the policy’s performance cannot be evaluated because we wouldn’t know the true best arm. Thus, the evaluation can only be done empirically, where the true best arm is known. The learner also would not know whether the problem he is facing is stationary or non-stationary, thus, a well-rounded policy is needed to tackle either stationary or non-stationary condition in MAB problem. Therefore, in this paper, multiple simulations were conducted on various conditions to test if Discounted Thompson Sampling policy was satisfactory to tackle stationary and piecewise-stationary condition. We used a metric called cumulative regret [6] to empirically evaluate the policy’s performance, this metric will be further explained in Section 2.3. Previously mentioned policies were also compared to A/B testing in terms of cumulative regret in the simulation that were conducted.

2. Method
This section consists of the workflow of Multi-Armed Bandit with Bernoulli Arms, the policies used and how it works, and simulation design.

2.1. Multi Armed Bandit workflow
MAB workflow has been briefly mentioned in Figure 2, this workflow is further explained in Figure 4 below.
The process in Figure 4 will be repeated until the experiments are over.

2.2 Policies used
This section will briefly mention the policies used in the simulation. Those policies are Epsilon First [14], Epsilon Greedy [2], SoftMax [3], Upper Confidence Bound [4], Thompson Sampling [5], and Discounted Thompson Sampling [12].

2.2.1. Epsilon First policy
The Epsilon First policy [14] also known as A/B testing, consists of two-phases, pure exploration phase and pure exploitation phase. This policy could be summarized in the following three simple steps:
1. Assign proportion of exploration $\epsilon$.
2. At the first $\epsilon \times 100\%$ of the experiment, do a randomized trial, this phase is called pure exploration.
3. After the pure exploration phase, choose the arm with the highest reward, this phase is called pure exploitation.

2.2.2. Epsilon Greedy policy
The Epsilon Greedy policy [2] could be summarized in several simple steps:
1. Assign probability of exploration $\epsilon$.
2. At each round, do the following:
   a. With a probability of $1 - \epsilon$, choose the arm with the highest mean reward (exploitation)
   b. Else, choose an arm uniformly at random (exploration)

2.2.3. SoftMax policy
The SoftMax policy implements SoftMax distribution [3] (also known as Boltzmann distribution) to give probabilities of choosing each arm. These probabilities are proportional to the mean reward of each arm and are shown by Equation (1) below:

$$ P(c_t = a_i) = \frac{\hat{\mu}_i(t-1)}{\sum_{s=1}^{k} e^{\frac{\hat{\mu}_s(t-1)}{\eta}}}, \quad \eta \in (0, \infty), \quad i = 1, 2, 3, \ldots, k \quad t = 1, 2, 3, \ldots, T. $$

This policy could be summarized in several simple steps:
1. Assign temperature parameter $\eta \in (0, \infty)$, this parameter control how much the proportion of exploration or exploitation, smaller $\eta$ will lean toward exploitation (the probabilities of the arm
with highest mean reward will be high), while larger $\eta$ will lean toward exploration (the probabilities will be more spread out).

2. At each round, the probabilities of choosing the $i$th arm ($P(c_t = a_i)$) are shown by equation (1)

$$P(c_t = a_i) = \frac{\text{Number of times } a_i \text{ has been chosen}}{\text{Total number of times an arm has been chosen}}$$

This policy could be summarized in several simple steps:

1. Assign bonus factor $\alpha \in (0, \infty)$ that control how much bonus is given to each arm, rarely explored arm (the arm with lower value of $N_i(t-1)$) will receive larger bonus.
2. At each round. Calculate upper bound of mean reward for each arm as shown by equation (2), and then choose the arm with the highest upper bound.

2.2.4. Upper Confidence Bound policy

The Upper Confidence Bound policy [4] implements an upper bound for average reward, this upper bound are composed by the average reward plus some bonus that is based on how rarely the arm are explored, this upper bound are shown by equation (2) below:

$$I_i(t) = \mu_i(t-1) + \frac{\alpha \log(t)}{2N_i(t-1)}, \quad \alpha \in (0, \infty)$$

$$i = 1, 2, 3, ..., k$$

$$t = 1, 2, 3, ..., T.$$  \hspace{1cm} (2)

This policy could be summarized in several simple steps:

1. Assign bonus factor $\alpha \in (0, \infty)$ that control how much bonus are given to each arm, rarely explored arm (the arm with lower value of $N_i(t-1)$) will receive larger bonus.
2. At each round. Calculate upper bound of mean reward for each arm as shown by equation (2), and then choose the arm with the highest upper bound.

2.2.5. Thompson Sampling policy

The Thompson Sampling policy [5] implements Bayesian method to determine which arm to choose at each round. This policy could be summarized in several simple steps:

1. Assign sampling model and its appropriate prior distribution for each arm.
2. At each round, construct posterior based on historical rewards for each arm.
3. For each arm, take a random sample from the posterior distribution.
4. Choose the arm with the highest random sample for that round.

For Multi-Armed Bandit with Bernoulli Arms, a Bernoulli sampling model and Beta($\alpha, \beta$) prior distribution could be chosen for each arm, and the posterior at each round will have a Beta distribution as shown by Equation (3) below:

$$I_i(t) \sim \text{Beta} \left( \alpha + S_i(t-1), \beta + F_i(t-1) \right), i = 1, 2, 3, ..., k; t = 1, 2, 3, ..., T$$  \hspace{1cm} (3)

with $S_i(t-1)$ denotes the cumulative reward and $F_i(t-1)$ denotes the cumulative failure. In this paper, the prior used is Beta(1,1) or Uniform(0,1).

2.2.6. Discounted Thompson Sampling policy

The Discounted Thompson Sampling policy [12] uses the same concept as Thompson Sampling policy but implements a discount factor to give lower weight as the records become older. In Multi-Armed Bandit with Bernoulli Arms, if Bernoulli sampling model and Beta($\alpha, \beta$) prior are chosen, then the discount factor will be implemented on the cumulative reward and cumulative failure that are shown in the equation (4) and equation (5) below:

$$\tilde{S}_i(t) = \begin{cases} 
\gamma \tilde{S}_i(t-1) + r_{tc_t}, & c_t = a_i \\
\gamma \tilde{S}_i(t-1), & c_t \neq a_i 
\end{cases}, \quad \gamma \in (0,1)$$  \hspace{1cm} (4)

$$\tilde{F}_i(t) = \begin{cases} 
\gamma \tilde{F}_i(t-1) + (1 - r_{tc_t}), & c_t = a_i \\
\gamma \tilde{F}_i(t-1), & c_t \neq a_i 
\end{cases}$$  \hspace{1cm} (5)

Previously mentioned policies are recapped in the Figure 5 below.
Figure 5. Recap of the policies used in this paper. There are 3 arms: blue, yellow, and green.

2.3. Cumulative regret
This section will explain the metric that are used to compare each policies. Let \( r_c \) denotes the reward given by the arm chosen by the learner at round \( t \), and let \( r_c^* \) denotes the reward given by the optimal arm at round \( t \), then the regret at round \( t \) are defined by the difference between \( r_c \) and \( r_c^* \). This regret symbolizes the loss if the learner didn’t chose the true optimal arm, because the true optimal arm is not known in the implementation in real life, the metric of regret can only be used empirically. Cumulative regret at round \( t \) are defined by Equation (6) below:

\[
R_t = \sum_{k=1}^{t} r_c - r_c^*, \quad t = 1, 2, 3, ..., T.
\]  

2.4. Simulation design
All policies that are previously mentioned in section 2.2 was simulated on various environments. The focus in this simulation are the Discounted Thompson Sampling policy, other previously mentioned policies will be served as a comparison of the Discounted Thompson Sampling policy performance. In this paper, 4 Bernoulli environments will be defined, which consists of 2 stationary and 2 piecewise-stationary. The environment specifications are listed in table 1 and table 2 below. The probabilities are generated by random, the probabilities in environment A are sampled (without replacement) from the set \{0.05, 0.10, 0.15, ..., 0.95\} while the probabilities in environment B are sampled (without replacement) from the set \{0.05, 0.10, 0.15, ..., 0.50\}.

Table 1. Probability of reward on each arm on stationary environment.

| Probability of reward | Stationary environment A | Stationary environment B |
|-----------------------|--------------------------|--------------------------|
| Arm 1                 | 0.30                     | 0.35                     |
Arm 2 0.75 0.15
Arm 3 0.90 0.25
Arm 4 0.80 0.05
Arm 5 0.95 0.20

Table 2. Probability of reward on each arm for each time period on piecewise-stationary environment (the number in bold show the optimal arm in that time period).

|                | Piecewise-stationary environment A | Piecewise-stationary environment B |
|----------------|------------------------------------|------------------------------------|
| 0 – 1999       | 0.20                               | 0.20                               |
| 2000 – 4999    | 0.95                               | 0.50                               |
| 5000 – 5999    | 0.05                               | 0.35                               |
| 6000 – 10000   | 0.50                               | 0.15                               |
| 0 – 1999       | 0.50                               | 0.25                               |
| 2000 – 4999    | 0.70                               | 0.60                               |
| 5000 – 5999    | 0.65                               | 0.95                               |
| 6000 – 10000   | 0.10                               | 0.20                               |

The probabilities on both piecewise-stationary environments are further visualized in Figure 6.

Figure 6. Piecewise-stationary environment A (left) and B (right). Visualized using SMPyBandits Framework [15].

Figure 7. The simulation process
Each policy will be run in parallel, each run will consist of different horizons, 5,000 rounds for the stationary environment, and 10,000 rounds for the piecewise-stationary environment. Each run will be repeated 100 times, and the evaluation will be based on the cumulative regret on those 100 runs. This simulation was done with the use of Lillian Besson’s SMPyBandits framework [15]. These simulations are summarized in Figure 7.

3. Result and Discussion

In this section we present the results of our simulation. Table 3 summarizes the mean and 1 standard deviation of the cumulative regret on 5 policies tested: Epsilon First, Epsilon Greedy, SoftMax, Upper Confidence Bound, Thompson Sampling, and Discounted Thompson Sampling. For each of the policy, four environments were set, represented by different values of the corresponding parameters (\(\epsilon, \eta, \alpha, \) and \(\gamma\)).

Table 3. Mean \(\pm\) 1 standard deviation of cumulative regret on various policies on 4 different environments (lower value is better, the number in bold shows the optimal policy in that environment).

| Policy                        | Stationary | Piecewise-stationary |
|-------------------------------|------------|----------------------|
| Baseline – Random Guess (Full Uniform Exploration) | 1310 \(\pm\) 34.1 | 749 \(\pm\) 30.5 |
| Epsilon First (\(\epsilon = 0.3\)) | 601 \(\pm\) 115 | 227 \(\pm\) 36.4 |
| Epsilon First (\(\epsilon = 0.4\)) | 691 \(\pm\) 71.9 | 298 \(\pm\) 33 |
| Epsilon First (\(\epsilon = 0.5\)) | 794 \(\pm\) 71.9 | 372 \(\pm\) 30.9 |
| Epsilon First (\(\epsilon = 0.6\)) | 884 \(\pm\) 27.8 | 451 \(\pm\) 29.9 |
| Epsilon Greedy (\(\epsilon = 0.50\)) | 976 \(\pm\) 308 | 474 \(\pm\) 155 |
| Epsilon Greedy (\(\epsilon = 0.25\)) | 854 \(\pm\) 617 | 432 \(\pm\) 310 |
| Epsilon Greedy (\(\epsilon = 0.10\)) | 718 \(\pm\) 765 | 436 \(\pm\) 411 |
| Epsilon Greedy (\(\epsilon = 0.05\)) | 741 \(\pm\) 790 | 500 \(\pm\) 384 |
| SoftMax (\(\eta = 4\)) | 985 \(\pm\) 26.1 | 738 \(\pm\) 28 |
| SoftMax (\(\eta = 2\)) | 932 \(\pm\) 27.7 | 728 \(\pm\) 26.4 |
| SoftMax (\(\eta = 1\)) | 821 \(\pm\) 24.6 | 700 \(\pm\) 28.6 |
| SoftMax (\(\eta = 0.5\)) | 653 \(\pm\) 28.2 | 653 \(\pm\) 34 |
| UCB (\(\alpha = 8\)) | 249 \(\pm\) 19.3 | 293 \(\pm\) 41.2 |
| UCB (\(\alpha = 4\)) | 176 \(\pm\) 17.2 | 194 \(\pm\) 34.5 |
| UCB (\(\alpha = 2\)) | 117 \(\pm\) 18.8 | 125 \(\pm\) 42.3 |
| UCB (\(\alpha = 1\)) | 69.7 \(\pm\) 17.9 | 68.1 \(\pm\) 36.2 |
| Thompson Sampling | 16 \(\pm\) 17.3 | 45.8 \(\pm\) 36.5 |
| D-Thompson Sampling (\(\gamma = 0.9995\)) | 24.5 \(\pm\) 17.4 | 51 \(\pm\) 38 |
| D-Thompson Sampling (\(\gamma = 0.999\)) | 30.8 \(\pm\) 20.6 | 62 \(\pm\) 48.7 |
| D-Thompson Sampling (\(\gamma = 0.995\)) | 79 \(\pm\) 22.6 | 152 \(\pm\) 45.3 |
| D-Thompson Sampling (\(\gamma = 0.99\)) | 120 \(\pm\) 22 | 234 \(\pm\) 41.7 |

Based on Table 3, Discounted Thompson Sampling policy achieved remarkable result compared to other policies on piecewise-stationary environments (achieved the lowest cumulative regret compared to other policies), all the while still achieving quite an optimal result in stationary environments. Note that the result of UCB policy, Thompson Sampling policy, and Discounted Thompson Sampling policy, far outclassed the result of Epsilon First policy (A/B testing). Furthermore, in piecewise-stationary environment B, A/B testing have a worse result than random guessing, this result has shown that hastily
deciding a ‘winner’ and only choosing the ‘winner’ without further exploration (pure exploitation) will bring bad result in a non-stationary environment.

Comparison of cumulative regret of the policies on defined piecewise-stationary environment are shown in Figure 8. Discounted Thompson Sampling policy achieved remarkable result compared to some well-known policies such as Thompson Sampling policy and UCB policy, and it is apparent that Discounted Thompson Sampling policy far surpassed A/B testing in minimizing regret.

**Figure 8.** Cumulative regret over the round (top) and histogram of cumulative regret at the end of the round (bottom) on piecewise-stationary environment A (left) and B (right).

### 4. Conclusion

Based on the empirical result of the simulations on various environments that have been defined. Discounted Thompson Sampling policy shows a quite remarkable and promising result in tackling piecewise-stationary conditions. Even on stationary conditions, Discounted Thompson Sampling policy still shows a quite good result, so it is a good choice of policy when the condition of the problem faced are unknown whether it is stationary or not. Different parameterization of Beta prior can be set to further see the performance of Discounted Thompson Sampling policy.

### Acknowledgments

This research was funded by Directorate of Research and Development of Universitas Indonesia (DRPM UI) as a grant of Publikasi Terindeks Internasional (PUTI) Prosiding 2020 No. NKB-977/UN2.RST/HKP.05.00/2020. Authors wishing to acknowledge assistance or encouragement from colleagues, special work by technical staff, and financial support from Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Indonesia.

### References

[1] White J Bandit 2012 Algorithms for website optimization O’Reilly Media Inc. 2012 Dec 24
[2] Sutton R S, Barto A G 2018 Reinforcement learning: An introduction MIT press 2018 Oct 19
[3] Cesa-Bianchi N, Gentile C, Lugosi G, Neu G Boltzmann 2017 Exploration done right In Advances in neural information processing systems p 6284-6293
[4] Auer P, Cesa-Bianchi N, Freund Y, Schapire R E 2002 The nonstochastic multiarmed bandit problem SIAM Journal on Computing 32(1) p 48-77
[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) p 285-94
[6] Slivkins A 2019 Introduction to multi-armed bandits. arXiv preprint arXiv:1904.07272
[7] Van Emden R, Kaptein M 2018 Contextual: Evaluating contextual multi-armed bandit problems in R. arXiv preprint arXiv:1811.01926
[8] Amat F, Chandrashekar A, Jebara T, Basilico J. 2018 Artwork personalization at Netflix. In *Proceedings of the 12th ACM conference on recommender systems* Sep 27 p 487-488
[9] Besbes O, Gur Y, Zeevi A 2014 Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in neural information processing systems* p 199-207
[10] Liu F, Lee J, Shroff N 2017 A change-detection based framework for piecewise-stationary multi-armed bandit problem arXiv preprint arXiv:1711.03539
[11] Gupta N, Granmo O C, Agrawala A. 2011 Thompson sampling for dynamic multi-armed bandits. In *10th International Conference on Machine Learning and Applications and Workshops* 2011 Dec 18 p 484-489 IEEE
[12] Raj V, Kalyani S. 2017 Taming non-stationary bandits: A Bayesian approach. arXiv preprint arXiv:1707.09727 2017 Jul 31
[13] Chapelle O, Li L. 2011 An empirical evaluation of thompson sampling. In *Advances in neural information processing systems* p 2249-2257
[14] Tran-Thanh L, Chapman A, Munoz De Cote Flores Luna JE, Rogers A, Jennings NR. *Epsilon–first policies for budget–limited multi-armed bandits*
[15] Besson L 2018 SMPyBandits: an Open-Source Research Framework for Single and Multi-Players Multi-Arms Bandits (MAB) Algorithms in Python