Thompson Sampling: An Increasingly Significant Solution to the Multi-armed Bandit Problem

Ziqi Shi[1, †], Linjie Zhu[2, *, †], Yiwei Zhu[3, †]

[1]Jinan Foreign Language School, Jinan, Shandong province 250000, China

[2]Shinyway Overseas Pathway College, Hangzhou, Zhejiang province 310005, China

[3]Qingdao No.58 middle school, Qingdao, Shandong province 266199, China

*Corresponding author email: 18403090@masu.edu.cn

[†]These authors contributed equally

Keywords: multi-armed bandit, Thompson sampling, ε-greedy, upper confidence bounds.

Abstract: This paper discusses the way to strengthen the probability and accuracy of multi-armed bandit. We are aimed at improving accuracy and probability when the reward of the problem is binary. We derive ε-Greedy and Upper Confidence Bounds algorithms in solving the multi-armed bandit problems and highlight the derivation and advantages of a most recent solution, namely the Thompson Sampling algorithms. Some researchers have proved that the Thompson Sampling is one of the better ways to solve the MAB problem. We show several applications of Thompson Sampling across varies of research fields to show how each problem is formulated and modeled. This paper helps people understand, review and strengthen the basic application and algorithm of Thompson Sampling and related solutions.

1. Introduction

The implementation of the multi-armed bandit problem can help people make better decisions about going out and maximizing the benefits. Multi-armed bandit (MAB) problem [1] is an algorithm that helps decision-makers to make maximum benefit decisions over time in the face of uncertainty. The question, therefore, arises from the idea of whether a player in a series of slot machines should decide which machine he should play once, and how often he should try another machine.

Then, this paper will introduce the classification of problems about MAB that are mentioned indifference paper. According to the papers that we found, there are five types of problems with MAB. First of all, theoretical understanding of the MAB problem algorithm was quite limited at that time, so it is necessary to show that the Thompson Sampling algorithm [2] achieves logarithmic expected regret for the stochastic multi-armed bandit problem, which is also the first problem that we faced. The paper Shipra Agrawal et. al. [2] proved it for the first time 5. After that, to achieve the maximum efficiency when we use MAB to solve some problems, many feedback control techniques usually need accurate models of the dynamics of the robot and its interaction with the surrounding environment, but most of the time, it is impossible to make it. The second problem is finding the most efficient way to make MAB more efficient. For example, in the paper Yahyaa S. et. al. [3], the main goal is finding the Pareto Front which is a set of optimal arms using the Pareto dominance relationship. The third one is using MAB to solve some problems in real life. The paper Li O. C. et. al. [4] had discussed how to use algorithms to solve exploration/exploitation or bandit problems. Not only this, in some papers, finding new algorithms based on the MAB was also becoming another topic and this is the fourth problem that we identified. For instance, the paper Munos R. et. al. [5] had already considered a variant of the basic algorithm of MAB problem which considered empirical variance of the different arms. Finally, some papers had also introduced some variants of the multi-armed bandit model and tried to deal with them. One of the examples is that the paper Canada's Michael Smith Genome Science Centre et. al. [6] introduced a new model which is a multi-armed bandit problem

with the known trend and in this new model the gambler knows the shape of reward function of each arm, but they do not know its distribution. This new problem is derived from many problems which are the online such as music, active learning, and interface recommendation applications. When the arm is sampled by the model, the reward which was received will change according to the trend that was known.

There is a lot of solutions for MAB. First is ε-Greedy [7] which is a method that chooses a random arm in each round to explore. If the probability of result is smaller than probability which is chosen by people, then, choosing another one until come out the highest empirical mean. The second is the Naive selection algorithm which is a method in that many a time experiments are performed on each handle, and the handle with the highest average return is selected. The third is Thompson sampling, supposing every machine has been explored many times. Then make a bata distribution, randomly pick a point from it, compare their lateral coordinate, and get the highest value. Thompson Sampling is more rigorous and accurate. It has more data, and data is more random. Other approaches are more limited. Like epsilon greedy just considers short term interest and naive selection algorithm waste a lot of time.

The MAB question can be used in many parts which have not been considered. Primary, in the beauty market, sales have many samples for customers to use. Customers will only answer good or bad. Sales do not know which is the best product, so they need to find out the best product. Secondly, the MAB problem is also popular in the Internet industry. In the delivery of advertisements, programmers need to optimize coding based on the number of clicks people have on ads to maximize profits and so on.

Thompson sampling is one of the parts of MAB problem. It is a very effective heuristic for solving the exploration/exploitation trade-off. In exploration, we take some risks to gather information about unknown options. In any case, Thompson sampling is easy to implement and should therefore be considered as a standard baseline. The purpose of the literature review is simple: the research and solution of this problem can have a helpful and progressive impact on the decision issues in finance, management, media, etc. The enrichment of this problem is an enrichment of unknown information in two major fields such as statistics and machine learning--researchers use a combination of programming techniques and statistical knowledge to solve real-life application problems - for example, applying programming to solve practical problems using UCB (upper confidence bound) [8, 9].

This article makes three main contributions. First, we provide a detailed review and summary of the solution to the MAB—an algorithm that helps decision-makers make decisions that maximize returns in the face of uncertainty. Secondly, Thompson's sampling is used as the focus of discussion for a review of his multifaceted applications - including Mechanical improvements, Online Influence, Online learning, Telecommunication, etc. The downside of pure Thompson Sampling is that it is fundamentally a regret-minimization algorithm.

2. The solutions of MAB

2.1 MAB formulation

Presuming every machine as an arm, player is user or anything which is controlling the machine, and reward is result (only be used when a question has only two possible outcomes it can be described by the Bernoulli distribution; for instance, yes or no, win or lose, good or bad, like 1, 0).

2.2 ε-Greedy Algorithm

To solve the MAB problem, there are three solutions. The first solution is the ε-Greedy Algorithm. This algorithm means that when we try to solve a problem, we will always make the best choice for the moment. In other words, through this algorithm, we get a locally optimal solution in a certain sense. However, the ε-Greedy Algorithm has a high risk when selecting a sub-optimal socket and, after that, stick to selecting. As a result, we just could not find the best socket. The ε-Greedy is probably the strategy that is the simplest and the most widely used to solve MAB problem and it was

first described by Watkins [10]. To solve MAB problems, what we need to do first is to explore with the probability ε, which means that we just choose one machine from N and let players try with the probability ε\N. Update the probability from 1 to N of the machine according to the feedback from the players. After that, we choose the machine which has the highest probability of getting the reward. In this way, exploration is added to the standard Greedy algorithm. Each action will be sampled repeatedly to get a more and more estimate of the true value. And because of the random sampling of action, the estimated reward values of all actions will converge on the true values. At this time, the disadvantages are shown. These non-optimal actions will be chosen continually, then long after they have been regarded as the non-optimal actions, their reward estimates will be refined. Because of this, the exploitation of the optimal actions is not maximized and the total reward will be less than it may get.

2.3 Upper Confidence Bounds

UCB is a deterministic algorithm for reinforcement learning that explores and develops confidence bounds based on the confidence bounds that the algorithm assigns to each machine in each round of exploration. These bounds are reduced when a machine is used more than others - when a person chooses an item to buy, if an item has been recommended k times (k feedbacks obtained), we can work out the probability that the item is good as

$$\tilde{p} = \frac{\sum reward}{k} \tag{1}$$

, $\tilde{P}$ is close to the true value as k approaches infinity, but in practice, the experimental data cannot be infinite, so our inferred probability and the actual probability will be a difference Δ, i.e.

$$\tilde{p} - \Delta \leq p \leq \tilde{p} + \Delta \tag{2}$$

So, people can define a new strategy: always be optimistic that the return on each item is$\tilde{p} + \Delta$ for each recommendation, which is the famous Upper Confidence Bound algorithm. In summary, there are three effective ways to obtain upper confidence bounds in the Multi-Armed Bandit Problem, namely Hoeffding's Inequality, UCB1, and Bayesian UCB.

2.3.1 Hoeffding's Inequality

Hoeffdings's inequality applies to bounded random variables. With a series of two independent random variables $X_1, \ldots, X_n \ldots$ Suppose that for all$1 \leq i \leq X_i$, x_i is an almost bounded variable, i.e., satisfies:$\mathbb{P} = 1$.

Seila contributes by reviewing some advanced aspects of approaches for assessing data generated by simulations [11]. Let $\{X_i, 1 \leq i \leq n\}$ be a negatively associated sequence, and let $\{X^*_i, 1 \leq i \leq n\}$ be a sequence of independent random variables such that X^*_i and X_i have the same distribution for each i=1, 2, ..., n. It is shown in this paper that

$$Ef\left(\sum_{i=1}^{n} X_i\right) \leq Ef\left(\sum_{i=1}^{n} X^*_i\right) \tag{3}$$

for any convex function f on R^1 and that

$$Ef\left(\max_{1\leq k\leq n} \sum_{i=k}^{n} X_i\right) \leq Ef\left(\max_{1\leq k\leq n} \sum_{i=1}^{k} X^*_i\right) \tag{4}$$

for any increasing convex function. [12]. Copulas and quasi-copulas are used by for similar possible best bounds on arbitrary sets of bivariate distribution functions with given margins.

Shivaswamy proposes a new boosting method based on a recently introduced notion called sample variance penalization, which is driven by an empirical version of Bernstein's inequality [13]. The goal of Matusz is to ensure a reliable and interpretable error bound, not to improve accuracy [14]. To this purpose, Pasargadae proposes the Fast-Hoeffding Drift Detection Method (FHDDM), which uses a sliding window and Hoeffding's inequality to find drift spots. Bentkus, Sason, and Bardenet are some of the other influential works [15-17].

2.3.2 UCB1

The UCB1 algorithm relies on a function that transforms a collection of average rewards from trial t into a set of decision values, which are then used to determine which goods to buy.

We want to create more confidence bound estimates when more rewards are observed, hence one heuristic is to reduce the threshold p in time.

Setting $p = t^{-4}$, as the number of t rounds increases, this small probability converges rapidly to 0. Eventually we obtain the UCB1 algorithm, which acts as: Here the previous equation here refers to the average observed reward of arm a at moment t, t is the current time step in the algorithm and n is the number of times arm a has been pulled so far.

$$U_t(a) = \sqrt{\frac{2logt}{N_t(a)}} \text{ and } a_t^{UCB1} = \underset{a \in \mathcal{A}}{argmax}\, Q(a) + \sqrt{\frac{2logt}{N_t(a)}} \tag{5}$$

2.3.3 Bayesian UCB

Bayesian-UCB is a unified framework for several variants of the UCB algorithm for solving different bandit problems (parametric multi-armed bandit problem, Gaussian bandit problem with unknown mean and variance, linear bandit problem).

When utilizing the frequentist cumulated regret as a measure of performance, Kaufmann shows that approaches derived from this second perspective perform well [18]. The goal of Russo is to provide a Bayesian regret bound for posterior sampling that can be tailored to a variety of model classes [19]. Maturana suggests that findings from this type of data have low translational potential for public health initiatives [20]. The goal of Kaufmann is to show that given a large class of prior distributions, if the distribution of rewards belongs to a one-dimensional family, the Bayesian algorithm UCB, which depends on the posterior distribution quantile, is asymptotically optimal [21]. Kirschner addresses bandits with heteroscedastic noise, where the noise distribution is explicitly allowed to depend on the evaluation point [22]. In order to improve the efficiency of the BO epoch, Dai proposes to combine BO in particular with Bayesian optimal stopping of the upper confidence limit of the Gaussian process [23]. TS-Cascade, a Thompson sampling algorithm for the cascading bandit issue, being investigated by Cheung [24].

Kharkovskii's Gaussian Process Upper Confidence Bound technology is the first privacy-preserving Bayesian Optimization solution with verifiable performance guarantees in an outsourcing environment. [25].

In the absence of a comparative study of compensation strategies, this article attempts to synthesize and highlight the influence of compensation strategies on the performance of the Bo algorithm on UCB through 11 numerical examples and two turbomachinery designs.

2.4 Thompson sampling

Thompson sampling, named after William R. The first time it come out is in 1933. Then it always is used for multi-armed problems. Thompson sampling is one of the oldest heuristics to address the exploration/exploitation trade-off, but it is surprisingly unpopular in the literature [26]. However, Thompson Sampling is not only about improving the estimate of the average reward and also enlarging the range of numbers that are chosen. This method is increasing accuracy and confidence. According to the data increasing, the Thompson Sampling will be more accurate. And the process of climbing people's trust is known as Bayesian Inference. When a question has only two possible outcomes it can be described by the Bernoulli distribution; for instance, yes or no, win or lose, good

or bad (like 1, 0). When the conclusion of the question is binary, then the beta distribution is the best way to have a model. The unknown number in the beta distribution is a sum. The formulation of a mean of the beta distribution is (α is number of successes, β is number of fails):

$$E[\beta] = \frac{\alpha}{\alpha+\beta} = \frac{numberofsuccesses}{totalnumberoftrials} \tag{6}$$

Formulation of Density of beta distribution:

$$f(x;\alpha,\beta) = constant \cdot x^{\alpha-1}(1-x)^{\beta-1}$$
$$= \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du} \tag{7}$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{8}$$

$$= \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{9}$$

The principle behind Thompson sampling is the Beta distribution mentioned above. Using the beta distribution, the Thompson Sampling can be used in four steps. First, take out the data α and β correspond to each group. Then, using α and β as parameters from different groups to produce a random number by beta distribution. After that, Sorting by random number and seeing data belong to which group. In the end, observing user feedback, if the user clicks, add 1 to the α, otherwise, add 1 to β. Why the Thompson Sampling is effective? At first, if a sample is selected a lot of times which means $\alpha+\beta$ is large, the distribution will be narrow. Use it to generate random numbers, close to the average. Secondly, if the data is not only $\alpha+\beta$ large (the distribution is narrow) but also $\alpha+\beta$ is also large and close to 1, meaning that this is good data, the average return is good. Finally, if an $\alpha+\beta$ of distribution is small and wide which means it has not been selected too many times, indicating that the date is not sure (This time, it is good, next time may be bad). But there is still a chance to exist, not completely abandoned.

3. The applications of Thompson sampling

3.1 Main content

In this part, we consider mechanical improvements (strengthening Vibration-Based Indoor Human Sensing Quality), Online Influence (Maximization under Independent Cascade Model with Semi-Bandit Feedback), telecommunication (A multi-armed bandit model for wireless network selection), online learning (Model-Independent Online Learning for Influence Maximization).

3.2 Mechanical improvements

Thompson sampling [2] was used in strengthening Vibration-Based Indoor Human Sensing Quality.[27] More and more people choose to install smart furniture. But due to the structure, layout and materials of the houses, the information collected by the sensor may be inaccurate. Therefore, the team of this paper used Thompson sampling [2] to strengthen the install smart furniture. They used sensor as an arm in MAB, user as a player in MAB, reward in this paper is the feedback from the user (good or bad).

Regarding how the data is collected, the first is that machine gets the possible sensor locations (provided by the CPS/IoT sensing system) and the deployment environment characteristics for each sensor location. Then, through the model, the machine recommends the best sensor location from the possible locations. At last, the users give feedback and the programmers update the model accordingly. They formulate this task as a MAB problem. They ran seven experiments with four machines in five

environments, each running 200 times. They use Normal distribution and Bayes' theorem to calculate the mean and the covariance matrix. And then based on that the system selects and recommends the location with the highest expected reward. The cause of improvement of Thompson sampling [2] is its better exploration and utilization of policies without tuning hyper-parameters and its parametric learning of location features with better generalization ability.As shown in table 1, it can clearly show that Thompson Sampling is more precise than the Random Selection in the early time steps which can attract the interest of users to do the survey (T is period).

Table 1. Evaluation in the offline setting. □@□ is short for □□□□□□@□. [27]

	T = 20			T = 50			T = 75			T = 100		
	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3
Thompson Sampling	47.89%	57.71%	91.82%	53.79%	59.21%	90.79%	55.11%	60.25%	90.04%	56.07%	60.43%	90.14 %
Random Selection	25.29%	48.89%	74.43%	24.43%	50.36%	75.25%	25.79%	49.71%	75.29%	24.39%	49.90%	74.71%

3.3 Online Influence

The MAB problem can be extended to maximise the impact of social networks, i.e., to maximise the number of users who are aware of the product to the 'initial' user base to which the product is exposed. Previous work assumed a model of known broadcast, but proposed a new parameterization method, using semi-bandit, to make the author frame unknown to the base broadcast and statistical power for data. When considering learning how to choose good seeds online, a new marketer wants to use the existing network to market their product. They need to select a good seed set while learning about the factors that influence the spread of information, which inspires the learning framework of the IM semi-bandit, working with marketers performing instant messaging in multiple 'rounds' and learning the factors that control the spread dynamically. In the influence-maximizing semi-robust problem, the agent knows $\mathcal{G}$ and $\mathcal{C}$, but not the diffusion model $\mathcal{D}$. Specifically, the agent does not know the model of $\mathcal{D}$. Consider a scenario in which the agent interacts with the social network in a T poll. In each round of $t \in \{1, \dots, T\}$, the agent first selects a seed set $\mathcal{S}_t \in \mathcal{C}$ based on its prior knowledge and past observations, and then naturally samples a diffusion random vector $\boldsymbol{w}_t \sim \mathbb{P}$. According to $\mathcal{D}(\boldsymbol{w}_t)$, influence spreads from $\mathcal{S}_t$ into the social network. After each such IM attempt, the agent observes the paired impact feedback and uses it to improve subsequent IM attempts. Each round corresponds to one IM attempt for the same or similar products. Each attempt results in a loss of impact diffusion (measured in terms of cumulative regret). This leads to the classic exploration-exploitation trade-off, where marketers either select seeds that will improve their knowledge of the diffusion process ('exploration') or find the set of seeds that will lead to the greatly expected diffusion ('exploitation'). To the end, a two-by-two influence semi-robust feedback model is proposed anlinuxnucb-based robber algorithm is developed. Independent analysis of our model shows that the regret bound is better depending on the size of the network, and experimentally evaluates our framework as a robust base diffusion model and can effectively learn near-optimal solutions [28].

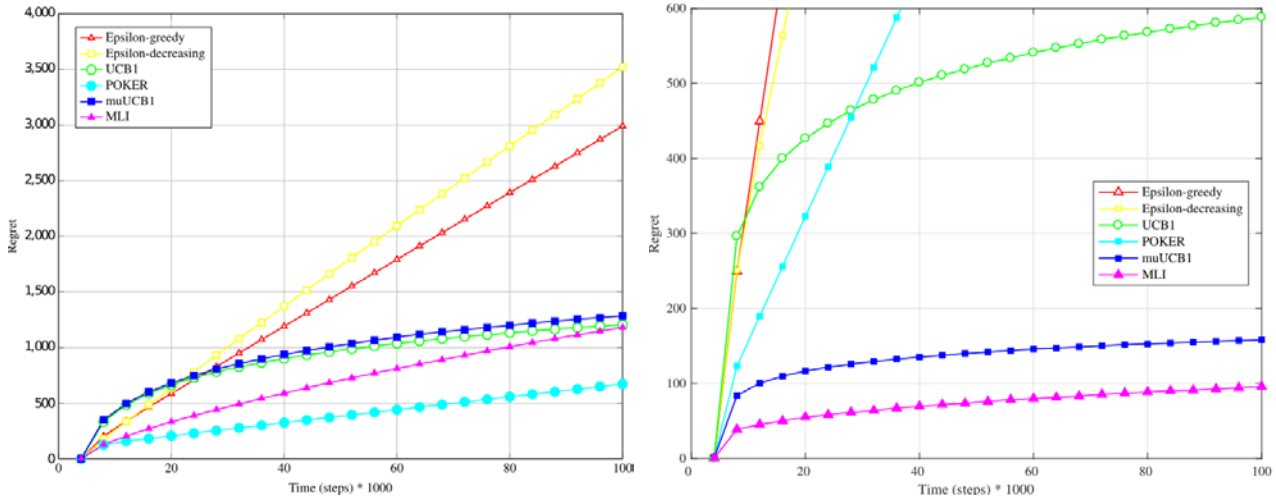
3.4 Online learning

Another notably influential article on the bandit problem in marketing fields such as media article [29] provides new ideas on the bandit problem - an analysis of IMLinUCB, a computationally efficient algorithm based on UCB. IMLinUCB represents its past observations as a positive definite matrix (Gram matrix) $\boldsymbol{M}_t \in \Re^{d \times d}$ and a vector$B_t \in \Re^d$.$\boldsymbol{M}_t = \boldsymbol{I} + \sigma^{-2}\boldsymbol{X}_t^{\top}\boldsymbol{X}_t$, $B_t = \boldsymbol{X}_t^{\top}Y_t$. (Let Xt be the matrix whose row is the eigenvector of all observed edges in step t, and Yt be the binary column vector in step t.) It receives the edge semi-robust feedback and uses it to update Mt and Bt. where, at each round t, the computational complexity of steps 1 and 3 of IMLinUCB are both $\mathcal{O}(|\mathcal{E}|d^2)$, it is of value1 that IMLinUCB reduces to CUCB, so that in some sense the confidence radius of IMLinUCB is the same as CUCB up to a logarithmic factor. This article also addresses the issue of influence maximization (IM), mainly in relation to social media campaigns and online learning, and offers new ideas to address IM. On many social networks, the probability of activation (of a user

being influenced) is unknown. One possibility is to learn this information from past propagation data. In practice, however, this data is difficult to obtain, and a large number of parameters makes learning challenging. This inspired the IM Bandit learning framework. Their cumulative regret bound is polynomial overall quantities of interest, achieving an approximate optimal dependence on the number of interactions and reflecting the topology of the network and the activation probabilities of its edges, thus providing insight into the complexity of the problem. This paper addresses these two challenges under the IC model of access-edge semi-robber feedback. Zheng Wen refers to their model as an independent cascaded semi-robber (ICSB). A new model-independent parameterisation and a corresponding agent objective function have been developed in response to the needs of the IM problem. We use this parameterisation to propose DILinUCB - a semi-robust learning algorithm for IM that does not rely on diffusion. The authors conjecture that a more statistically efficient algorithm with an additional O(Vn) factor removed from the regret bound may be obtained by a proper generalisation of the source node.

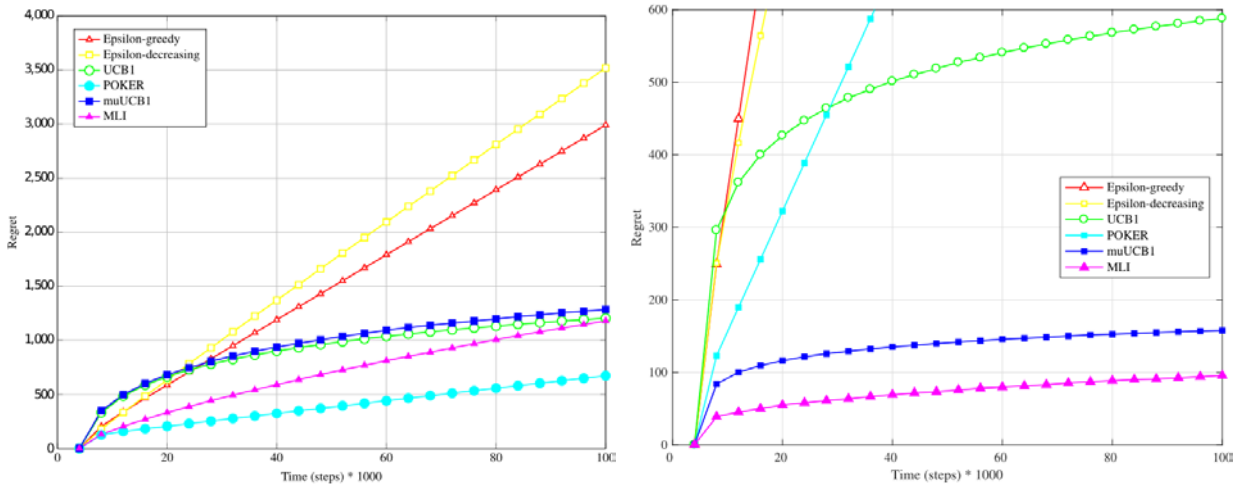
3.5 Telecommunication

One of the fields of the applications of MAB and Thompson sampling that this part introduces is telecommunication. This application came from the paper Stefano Boldrini et. al. [30]. This problem can be defined as “Which wireless network that is available to use could offer the best performance that has the best quality to the users”. So, the main goal is to maximize the quality that the final users could experience. At first, it just introduced a new model called muMAB which predicted two kinds of actions: measure and use. After that, the paper also introduced two other new algorithms: measure-use-UCB1(muUCB1) and Measure with Logarithmic Interval (MLI), which are based on the model muMAB, to analyze the effect of muMAB. Then they decided to have the tests on the impact of the introduction of the proposed model which was carried out through simulations and compared the regret of six different algorithms which are: UCB1 [31], muUCB1, MLI, ε-decreasing [31, 32], ε-greedy [33], POKER [34]. These six algorithms were tested against both synthetic and captured data which followed some common settings. The synthetic data were produced by the three different distributions of the reward probability density function, the Bernoulli distribution, the truncated Gaussian distribution and the exponential distribution, with different ratio of TU/TM. And there are two different configurations, which are the Hard configuration and the Easy configuration. Then they had the simulations for the data and used three ways to obtain the result and got an analysis of the effect of the logarithmic conversion. The first one was the synthetic rewards generated using the Hard configuration. For example, for the Bernoulli distribution, there are three figures.



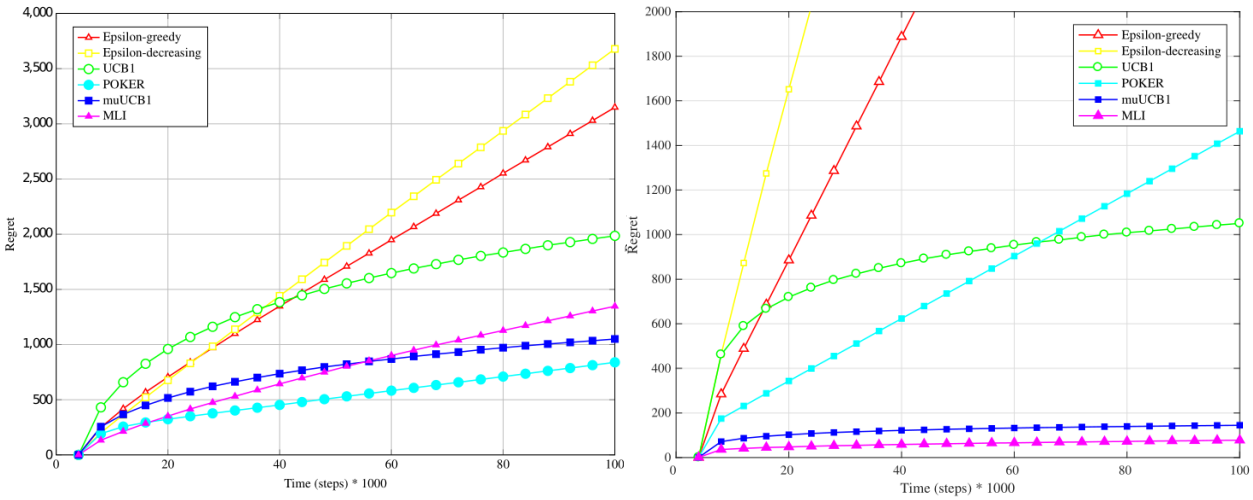
(a) Hard configuration (b) Easy configuration

Figure 1. Considered the performance of the regret of the six different algorithms and the probability density function is Bernoulli distribution when TU/TM=1.



(a) Hard configuration (b) Easy configuration

Figure 2. Considered the performance of the regret of the six different algorithms and the probability density function is Bernoulli distribution when TU/TM=5



(a) Hard configuration (b) Easy configuration

Figure 3. Considered the performance of the regret of the six different algorithms and the probability density function is Bernoulli distribution when TU/TM=10.

In this situation, the POKER algorithm gave the best overall behavior according to these three graphs. And could also find that for the new algorithms, their performance improved compared to the other algorithms. The muUCB1 algirithm had a bad performance when TU/TM=1, but when TU/TM=10, muUCB1 had a regret comparable to the POKER algorithm. The second result was obtained in the Easy configuration. Then the third one was obtained using real data and linear conversion. In the end, the paper Vermorel, J. et. al. [34] analyzed the results and found that the conservative algorithm will spend more time in measuring, such as UCB1 and muUCB1. It should be preferred when the different arms have similar rewards.

According to the result, the MLT algorithm and the muUCB1 algorithm could use the measuring phase that is introduced by the muUCB1 algorithm. As the increase of the TU/TM ratio, the performance of these two algorithms will continue to be improved. All in all, we can just choose the new model, which is muMAB, to describe the question of network selection, because it is more flexible than the traditional algorithm.

3.6 Discussion

The four applications introduced in this part are just a part of all applications of MAB problems and Thompson sampling, which shows that the Thompson sampling is widely used in various fields

in our life. But sometimes the variants of the Thompson sampling are more preferred than itself. For example, the fourth application uses the model muMAB and some other algorithms which are the variants of the MAB and the model IMLinUCB used in the third application about online learning is also the variant of UCB algorithms which is one of the solutions of Thompson sampling. These four applications also offer a new way to solve problems, which is that just turn the problems that are difficult to understand and solve into data that is more visualized and pellucid. So, it is necessary for the world to be digitized and the IT industry and the big data which are growing fast now are just suitable examples to verify this trend. Thompson sampling is also used in robotics [35], such as multi-target such scenarios [36, 37].

4. Conclusion

This article focuses on the multi-armed bandit problem and Thompson sampling, one of the solutions to the multi-armed bandit problem. Thompson sampling and the multi-armed bandit problem is a relatively mature problem, and a summary and discussion of the multi-armed bandit problem can also help us to make better decisions about the problem. In the past year, there have been some recent developments in the multi-armed bandit problem and Thompson sampling. For the multi-armed bandit problem, the agents' regret with respect to the optimal allocation is poly-logarithmic in the time horizon was smoothly demonstrated. Two algorithms that use a bandit to find the optimal exploration of the contextual bandit algorithm was proposed, which might be the first step towards the automation of the multi-armed bandit algorithm, and in general the research on the multi-armed bandit problem over the past year has led to better exploitation of the problem and its use by humans. The solution to this problem (the multi-armed bandit problem), the Thompson sampling method, has also made valuable progress and practical applications - the problem of recommending relevant contents to users of internet platforms in the form of lists of items, called "slates", can further refine this solution. In summary, both the multi-armed bandit problem and Thompson sampling are valuable and worthy of study, and this article will not only provide an understanding of the basics and applications of these two related topics but will also help one to review and review the multi-armed bandit problem and Thompson sampling.

References

[1] Vermorel, J., & Mohri, M. (2005, October). Multi-armed bandit algorithms and empirical evaluation. In: European conference on machine learning Springer, Berlin, Heidelberg. pp. 437 - 448.

[2] Agrawal, S., & Goyal, N. (2012, June). Analysis of thompson sampling for the multi-armed bandit problem. In: Conference on learning theory. JMLR Workshop and Conference Proceedings. pp. 39.1 - 39.26.

[3] Yahyaa, S. Q., Drugan, M. M., & Manderick, B. (2015, January). Thompson Sampling in the Adaptive Linear Scalarized Multi Objective Multi Armed Bandit. In: ICAART (2). pp. 55 - 65.

[4] Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. Advances in neural information processing systems, 24.

[5] Audibert, J. Y., Munos, R., & Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. Theoretical Computer Science, 410 (19): 1876 - 1902.

[6] Bouneffouf, D., & Féraud, R. (2016). Multi-armed bandit problem with known trend. Neurocomputing, 205: 16 - 21.

[7] Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction. Cambridge, MA: MIT Press.

[8] Lattimore, T., & Szepesvári, C. (2020). Bandit algorithms. Cambridge University Press.

[9] Elmasry, G. F. (2020). IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and Other Advanced Radio Communication Systems.

[10] Watkins, C. J. C. H. (1989). Learning form delayed rewards. Ph. D. thesis, King's College, University of Cambridge.

[11] Seila, A. F. (1992, December). Advanced output analysis for simulation. In: Proceedings of the 24th conference on Winter simulation. pp. 190 - 197.

[12] Shao, Q. M. (2000). A comparison theorem on moment inequalities between negatively associated and independent random variables. Journal of Theoretical Probability, 13 (2): 343 - 356.

[13] Shivaswamy, P., & Jebara, T. (2010, March). Empirical bernstein boosting. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings. pp. 733 - 740.

[14] Matuszyk, P., Krempl, G., & Spiliopoulou, M. (2013, October). Correcting the usage of the hoeffding inequality in stream mining. In: International Symposium on Intelligent Data Analysis. Springer, Berlin, Heidelberg. pp. 298 - 309.

[15] Bentkus, V. (2008). An extension of the Hoeffding inequality to unbounded random variables. Lithuanian Mathematical Journal, 48 (2): 137 - 157.

[16] Sason, I. (2011). On refined versions of the Azuma-Hoeffding inequality with applications in information theory. arXiv preprint arXiv:1111.1977.

[17] Bardenet, R., & Maillard, O. A. (2015). Concentration inequalities for sampling without replacement. Bernoulli, 21 (3): 1361 - 1385.

[18] Kaufmann, E., Cappé, O., & Garivier, A. (2012, March). On Bayesian upper confidence bounds for bandit problems. In: Artificial intelligence and statistics. PMLR. pp. 592 - 600.

[19] Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. Mathematics of Operations Research, 39 (4): 1221 - 1243.

[20] De Maturana, E. L., Chanok, S. J., Picornell, A. C., Rothman, N., Herranz, J., Calle, M. L., ... & Malats, N. (2014). Whole genome prediction of bladder cancer risk with the Bayesian LASSO. Genetic epidemiology, 38 (5): 467 - 476.

[21] Kaufmann, E. (2018). On Bayesian index policies for sequential resource allocation. The Annals of Statistics, 46 (2): 842 - 865.

[22] Kirschner, J., & Krause, A. (2018, July). Information directed sampling and bandits with heteroscedastic noise. In: Conference on Learning Theory. PMLR. pp. 358 - 384.

[23] Dai, Z., Yu, H., Low, B. K. H., & Jaillet, P. (2019, May). Bayesian optimization meets Bayesian optimal stopping. In: International Conference on Machine Learning. PMLR. pp. 1496 - 1506.

[24] Cheung, W. C., Tan, V., & Zhong, Z. (2019, April). A Thompson sampling algorithm for cascading bandits. In: The 22nd International Conference on Artificial Intelligence and Statistics. PMLR. pp. 438 - 447.

[25] Kharkovskii, D., Dai, Z., & Low, B. K. H. (2020, November). Private outsourced Bayesian optimization. In: International Conference on Machine Learning. PMLR. pp. 5231 - 5242.

[26] Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. Advances in neural information processing systems, 24.

[27] Yu, T., Zhang, Y., Hu, Z., Xu, S., & Pan, S. (2021, May). Vibration-based indoor human sensing quality reinforcement via Thompson sampling. In: Proceedings of the First International Workshop on Cyber-Physical-Human System Design and Implementation. pp. 33 - 38.

[28] Vaswani, S., Kveton, B., Wen, Z., Ghavamzadeh, M., Lakshmanan, L. V., & Schmidt, M. (2017, July). Model-independent online learning for influence maximization. In: International Conference on Machine Learning. PMLR. pp. 3530 - 3539.

[29] Wen, Z., Kveton, B., Valko, M., & Vaswani, S. (2017). Online influence maximization under independent cascade model with semi-bandit feedback. Advances in neural information processing systems, 30.

[30] Boldrini, S., De Nardis, L., Caso, G., Le, M. T., Fiorina, J., & Di Benedetto, M. G. (2018). mumab: A multi-armed bandit model for wireless network selection. Algorithms, 11 (2): 13.

[31] Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47 (2): 235 - 256.

[32] Cesa-Bianchi, N., & Fischer, P. (1998, July). Finite-Time Regret Bounds for the Multiarmed Bandit Problem. In: ICML, Vol. 98, pp. 100 - 108.

[33] Watkins, C. J. C. H. (1989). Learning form delayed rewards. Ph. D. thesis, King's College, University of Cambridge.

[34] Vermorel, J., & Mohri, M. (2005, October). Multi-armed bandit algorithms and empirical evaluation. In: European conference on machine learning. Springer, Berlin, Heidelberg. pp. 437 - 448.

[35] Chen, J., Xie, Z., Dames, P. (2022) The semantic PHD filter for multi-class target tracking: From theory to practice. Robotics and Autonomous Systems, 149, 103947.

[36] Chen, J., Dames, P. (2020) Collision-free distributed multi-target tracking using teams of mobile robots with localization uncertainty. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 6968-6974). IEEE.

[37] Chen, J., Park. S (2022) Bernoulli Thompson Sampling-based Target Search Algorithm for Mobile Robots.