International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 02, 2021

Reinforcement Learning Based Empirical Comparison of UCB, Epsilon-Greedy, and Thompson Sampling

Anupam Singh

School of Computer Science, UPES, Dehradun, Uttarakhand India

Email: anupam.singh@ddn.upes.ac.in

Abstract: Reinforcement learning is a referred for what to do, how to align conditions to action, which leads to enhance analytical reward signal. The learner is not stated to determine which actions to take, although it must determine which actions return the most reward by applying them. In most exciting and puzzling cases, the actions may affect immediate reward and also the following situation and, through that, all succeeding rewards. These two characteristics as trial-and-error search and delayed reward are the two most significant distinguishing features of reinforcement learning. The main difference between reinforcement learning and other forms of learning is that it uses training knowledge to measure the actions performed rather than instructing by providing correct actions. This is what necessitates active exploration and an explicit quest for appropriate actions. Purely evaluative feedback shows how effective the action was, but not whether it was the best or worst choice. This paper investigates on a comparative analysis of Epsilon-Greedy, UCB, and Thompson sampling algorithm. Experimental results gives clear insight of comparative analysis of process.

Keywords: Learning, UCB, Epsilon Greedy, Thompson Sampling, Multi Armed Bandit

1. INTRODUCTION

"Learning in which we examine how an agent can learn from success and failure, from reward and punishment". The technique that we study the interaction with our environment is possibly the first to occur to us when we think about learning [1] nature.

1.1 Motivation

In past few years, field of learning and sampling information has seen some great developments due to communication technology's tremendous advancements and decision support system. Internet's rapid growth along with high dimensional data set and cluster technology's widespread use resulted for advanced sampling in opening up the new chances.

Purely instructive feedback, on the other hand, suggests the proper course of action regardless of the action taken. The foundation of supervised learning, which includes large parts of pattern classification, artificial neural networks, and system identification, is this type of feedback. These two types of feedback are very different in their purest forms: evaluative feedback is entirely contingent on the action taken, while instructive feedback is not.

The work tilted as "Comparative Analysis of Epsilon-Greedy [2, 3], UCB[4] and Thompson Sampling algorithms[5,6,7] in Reinforcement Learning" uses Multi-arm bandit problem to compare the performance of the three mentioned algorithms in a simple environment setting. The results showed that Upper-Confidence Bound and Thompson Sampling algorithms outperformed Epsilon-Greedy algorithm. Also, the time taken by the former two algorithms to reach their peak reward values was less than what it took for the Epsilon-Greedy algorithm to reach a peak value.
An infant has no explicit teacher when it plays, waves its arms, or looks around, but it does h ave a direct sensorimotor connection to its environment. Exercising this link yields a wealth o f information about cause and effect, the consequences of actions, and what to do to achieve o bjectivesSuch interactions are unquestionably a major source of knowledge about our environ ment and ourselves throughout our lives. We are acutely aware of how our environment resp onds to what we do, whether we are learning to drive a car or hold a conversation, and we see k to influence what happens through our behaviour. Learning from interaction is a fundamental concept that underpins nearly all learning and intelligence theories.

1.2 Our Contributions

In this study we explore three algorithms – Epsilon-Greedy [2, 3], Upper-Confidence Bound [4] and Thompson Sampling [5, 6, 8], on the famous multi-armed bandit problem. In doing so, we compare how the three algorithms differ in performance in a simple environment.

Consider the following example of a learning challenge. You are frequently confronted with a decision between k various alternatives or behaviors. Following each option, you will be given a numerical reward based on a stationary probability distribution that is dependent on the action you choose. Your goal is to maximize the total predicted reward over a set length of time, such as 1000 action options or time steps. This is the original version of the k-armed bandit issue, which is called after a slot machine or "one-armed bandit," but with k instead of one lever. Each move is similar to pulling a lever on a slot machine, and the prizes are the payouts for striking the jackpot. By concentrating your actions on the best levers, you can increase your rewards through repeated action selections. Another example is a doctor deciding between experimental therapies for a group of critically ill patients. Patient's wellbeing as a reward is based upon action of treatment.
To solve such problems, we need to define certain steps in order to choose the actions based on the state of the environment and the choices we have. Also, we need to optimize it in such a manner that we reach the maximum possible reward efficiently. For this purpose, we have done a comparative analysis to see how different algorithms perform with a simple environment setting.

1.3 Organization of work

The paper is organized as section 2 depicts the literature of the domain, section 3 focuses on problem statement. The work discussed in section 4 gives a clear insight of proposed solution, section 5 gives the flow of data, section 6 depicts the algorithm and at last section 7 concludes the paper

2. RELATED WORK

A Multi-Armed Bandit is a mathematical model that can be used to reason about how to make decisions when an agent (learner) have multiple actions it can take and imperfect information about the rewards it would receive after taking those actions. In comparison to Thompson Sampling and UCB-1, this work focuses primarily on the Epsilon Greedy Algorithm (Upper Confidence Bound). It discusses the advantages of employing bandit algorithms over A/B testing [9] and compares and contrasts the three primary methods. It demonstrates the Epsilon Greedy Algorithm's best use cases, which are when the experimentation duration is longer than A/B testing and you want to utilize the top performing variant. It also discusses situations where the algorithm fails to produce statistically valid findings, such as when the sample size for each path of the experiment is extremely tiny.

The work discussed in [10,11,12,13] depicts how learning can be enhanced and impact of UCB as well as sampling can be incorporated. Reinforcement learning plays a vital role in process of data training and utilization in decision making.

3. PROBLEM STATEMENT

Reinforcement learning is an aspect of machine learning which is uses observed reward to learn an optimal policy for the environment. MAB stands for Multi-Armed Bandit and is a type of reinforcement learning. An agent (learner) in a multi-armed bandit implementation chooses among k various uncertain actions and is rewarded based on the action chosen. Multiple different algorithms are available for solving the multi-armed bandit problem but each and every algorithm has its trade off. So, the exploitation-versus-exploration dilemma in reinforcement learning turns out to be of huge consideration while working with the multi-armed bandit problem.

The objective is to compare three different strategies that deal with the multi-armed bandit's exploitation-versus-exploration dilemma: the epsilon-greedy strategy, upper confidence bound interval strategy, and Thompson sampling. In doing so, we compare how these algorithms differ in performance in a simple environment. Additionally, the performances of the three mentioned algorithms over long and short step sizes have been stated and analyzed.

4. PROPOSED SOLUTION

The methodology to determine the best approach to solve the multi-armed bandit [9,14,15] problem initiates by the understanding of A/B Testing[12], defining the environment in which the algorithms will run, and the application of the algorithms used to solve the multi-armed bandit problem.

4.1 A/B Testing

The aim of A/B testing [12] is to estimate the variant of an experiment that is really more effective than another. While performing an A/B experiment, users are divided into 2 groups namely the treatment group and the control group. The treatment group is given access to the one variant while the control group has access to another variant. The goal of the A/B is then to compare the conversion rates of the two groups using statistical inference.

4.2 Environment Setup

An environment with a pre-defined number of bandits has been created. All bandits behave randomly, but on average, each of them returns a specified profit. Some bandits have higher returns compared to others and the aim is to maximize the profit. To interact with the environment, one needs to specify the bandit they choose to pull, and the environment returns a reward for that action.

4.3 Application of Algorithms

In reinforcement learning, the agent tries to determine its environment. The agent firstly has none or partial knowledge related to environment. The agent may selects to discover by selecting an action with an unidentified outcome, to get more data about the environment. Alternatively, it can choose to exploit and select an action based on its previous knowledge of the environment to get a decent reward.

The exploitation-versus-exploration dilemma occurs every time an agent wants to act as optimally as possible but does not know which action is optimal because it does not have knowledge of which action is best for it.

After understanding the exploitation-versus-exploration dilemma, the following algorithms will be applied.

4.3.1 Epsilon-Greedy

The epsilon-greedy [2,3] algorithm begins by setting epsilon to a low value. After that, a random probability value between 0.0 and 1.0 is generated for each trial. A random arm is chosen if the produced probability is less than (epsilon). Otherwise, the arm with the highest average reward at the time is chosen.

4.3.2. Upper Confidence Bound

"Optimism in the face of uncertainty" is how the UCB1 [4] algorithm is described. The UCB1 algorithm creates a "upper confidence bound" based on a history of rewards and the number of pulls for each arm, which it uses to determine which arm to pull. Despite being unclear about the estimated means of the distributions of the k arms, this algorithm always selects the arm I that maximizes the sum x I +a(i,t), where x I is the estimated mean of the machine I and a(i,t) is the UCB of machine I at time t [4].

The upper confidence bound of the machine *i* at time step *t* is

$$a(i,t)=\sqrt{2\log(t) / n_j}$$

Where n_j is the number of times the machine *i* have been pulled at timestep *t*.

4.3.3. Thompson sampling

Thompson Sampling [5, 6, 16] is a probabilistic algorithm based on Bayesian ideas. The algorithm assumes that all machines have a uniform distribution of the probability of success, in this case getting a reward. For each observation obtained from a machine, based on the reward a new distribution is generated with probabilities of success for each machine. Further observations are made based on these prior probabilities obtained on each round or observation which then updates the success distributions. The work presented in[17] depicts how the problem of wheat diseases can be resolved through machine learning. After sufficient observations, each machine will have a success distribution associated with it which can helps in choosing the machines wisely to get the maximum rewards [18,19].

5. PSEUDO CODE

5.1. Class **Epsilon_Bandit**

i. __init__(m)

This function acts as a constructor. 'true_mean' is set to 'm', 'estimated_mean' and N are set to 0.

ii. pull( )

This function is used to pull a sample from the real data distribution of the bandit, modeled as a Gaussian centered as 'true_mean'.

iii. update(x)

This function is used to update estimated mean using new sample pulled from real distribution.

5.2. Class **Confidence_Interval_Bandit**

i. __init__(m)

This function acts as a constructor. 'true_mean' is set to 'm', 'estimated_mean' is set to 0 and N is set to a small non-zero value.

ii. pull( )

This function is used to pull a sample from the real data distribution of the bandit, modeled as a Gaussian centered as 'true_mean'.

iii. update(x)

This function is used to update estimated mean using new sample pulled from real distribution.

5.3. Class **Bayesian_Bandit**

i. __init__(m)

This function acts as a constructor. 'true_mean' is set to 'm', 'estimated_mean' is set to 0, 'estimated_precision' is set to 1, 'tau' is set to 1 and 'sum_of_x' is set to 0.

ii. pull( )

This function is used to pull a sample from the real data distribution of the bandit, modeled as a Gaussian centered as 'true_mean'.

iii. sample( )

This function returns a sample from the estimated distribution.

iv. update(x)

This function updates estimated distribution using new sample pulled from the real distribution.

5.4. run_experiment_fixed_epsilon(m1, m2, m3, epsilon, N)

This function is used to run three epsilon bandits (m1, m2, m3) over N steps and return the cumulative average of the runs.

5.5. run_experiment_confidence_interval (m1, m2, m3, N)

This function is used to run three confidence interval bandits (m1, m2, m3) over N steps and return the cumulative average of the runs.

5.6. run_experiment_bayesian (m1, m2, m3, N)

This function is used to run three Bayesian bandits (m1, m2, m3) over N steps and return the cumulative average of the runs.

6. RESULT DISCUSSIONS

In this study, we concluded that over a longer period, UCB and Thompson Sampling performed better than Epsilon-greedy(Figure 1). We also found that UCB and Thompson Sampling reach the maximum possible reward value much quicker than Epsilon-greedy algorithm.

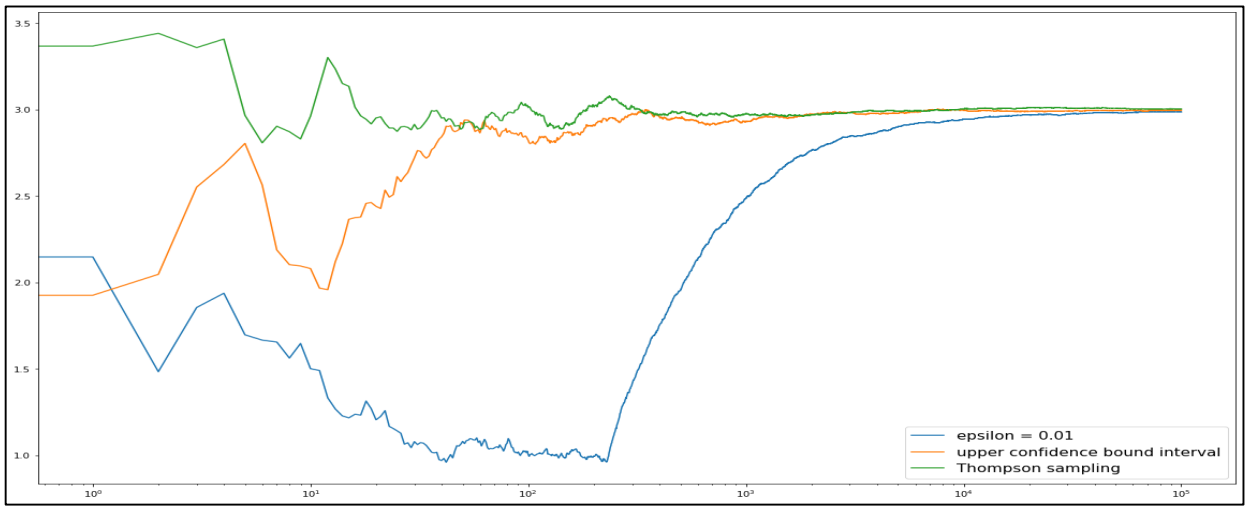

Fig. 1: Graph showing the performance over 100000 steps

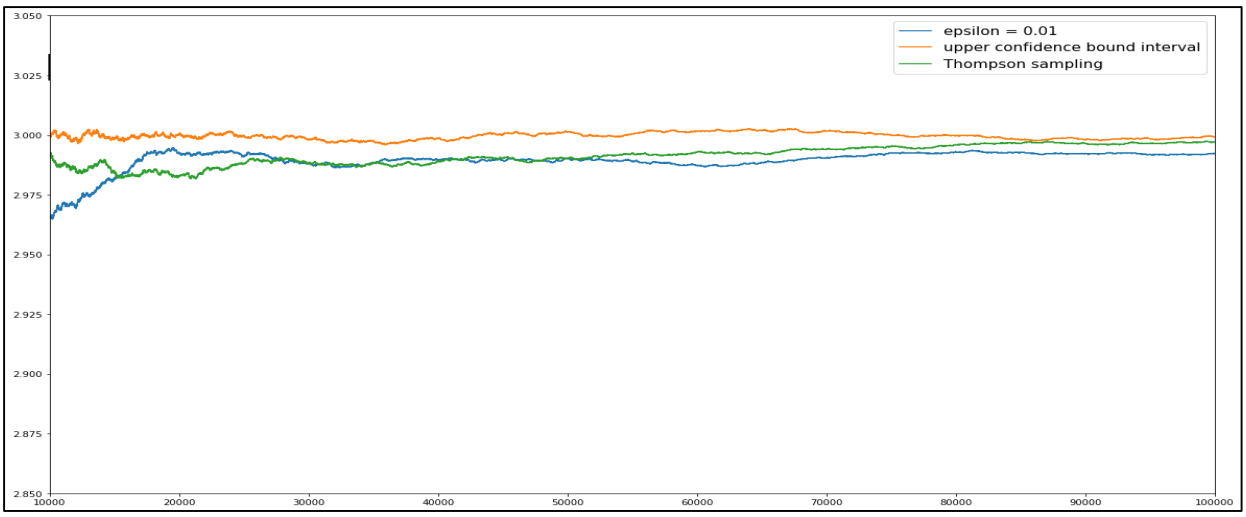

Fig. 2: Graph showing the performance over 100000 steps, closeup for the last 1000 steps

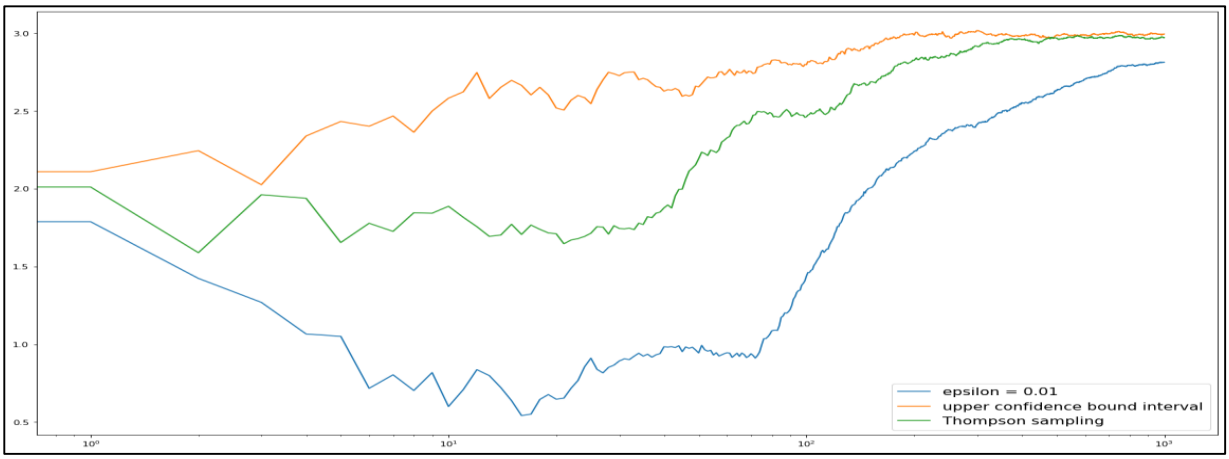

Fig. 3: Graph showing the performance over 1000 steps

In this study, we performed the comparative assessment of three usually used algorithm's namely Epsilon-Greedy, UCB and Thompson Sampling on MAB. The difference between MAB and A/B testing is that A/B testing only allows you to explore while the experiment is ongoing and exploit afterward (Figure 2, Figure 3). User can explore and exploit at the same time using MAB, so user don't have to wait for the experiment to finish before user start exploiting. As a result, MAB speeds up the experiment's completion and decreases total regret (decrease in potential rewards due to executing the learning algorithm instead of performing optimally from the start).

We observed, initially at the start of each algorithm, the pattern of each step up to 1000 steps are inconsistent. Among that the algorithm Thomson sampling and Epsilon-Greedy performance were found to be consistent over 10,000 steps. Whereas UCB performance was observed to be quiet more uniform than former algorithm's. Although, the result indicated that the studied algorithm near 1, 00,000.00 steps showed more consistent compared to the lower steps. These algorithm's falls under Reinforcement learning, which works in association with the environment where no or incomplete knowledge is available in which the agent tries to learn from its sequential decisions to maximize the output.

The Epsilon-greedy algorithm is simple to implement. It smoothly maintains the balance between exploitation and exploration and regarded as the greediest algorithm. It can generates results with minimum input of samples compared to Thomson sampling. Although, percent distribution of variation is fixed compared others. UCB-1 algorithm performance observed to be consistent over time and would not show any randomness. The most successful variation will continue to perform at the highest level. At most of the time when more data is accumulated the algorithm will exploits as opposed to explore. Although, this algorithm works with some concerns such as

The distribution percentage of a variant may approach zero depending on how the algorithm is running. Its performance may be reduced/loss in comparison to Thomson sampling and It requires expertise to implement.

Thomson algorithm is consists of Bayesian approach. It produce a vector of expected output for each arm from a subsequent distribution and then update it. It is one of the primary algorithm which provides higher accurate result even in cases where the difference in the payout rates of the two paths is very less. Although, high number of samples yield better significant result. In comparison to Epsilon-greedy algorithm it does not exploit more hence lesser cumulative reward and requires expertise to use it.

7. CONCLUSIONS

Upper-confidence bound and Thompson sampling algorithms turned out to be better than Epsilon-greedy algorithm over longer and shorter run times. The former two algorithms consistently reaped the maximum possible reward, and even achieved it faster than epsilon greedy algorithm. This clearly shows how tied-down epsilon-greedy algorithm is due to its probabilistic approach in choosing the best bandit per turn.

8. REFERENCES

[1] Richard S. Sutton and Andrew G. Barto, Cambridge MA : The MIT Press, *Reinforcement Learning : An Introduction*, 2018 .

[2] Raykar, Vikas, and Priyanka Agrawal. "Sequential crowdsourced labeling as an epsilon-greedy exploration in a Markov Decision Process." *Artificial intelligence and statistics*. PMLR, 2014.

[3] Riti Agarwal, International Journal of New Technology and Research (volume : 6, issue : 9th September 2020), *"The Epsilon Greedy Algorithm – A Performance Review"*, September 2020.

[4] Garivier, Aurélien, and Eric Moulines. "On upper-confidence bound policies for switching bandit problems." *International Conference on Algorithmic Learning Theory*. Springer, Berlin, Heidelberg, 2011.

[5] Riquelme, Carlos, George Tucker, and Jasper Snoek. "Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling." *arXiv preprint arXiv:1802.09127* (2018).

[6] Hernández-Lobato, José Miguel, et al. "Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space." *International conference on machine learning*. PMLR, 2017.

[7] Umami, Izzatul, and Lailia Rahmawati. "Comparing Epsilon Greedy and Thompson Sampling model for Multi-Armed Bandit algorithm on Marketing Dataset." *Journal of Applied Data Sciences* 2.2 (2021).

[8] Viappiani, Paolo. "Thompson sampling for bayesian bandits with resets." *International Conference on Algorithmic DecisionTheory*. Springer, Berlin, Heidelberg, 2013.

[9] Elena, Gangan, Kudus Milos, and Ilyushin Eugene. "Survey of multiarmed bandit algorithms applied to recommendation systems." *International Journal of Open Information Technologies* 9.4 (2021): 12-27.

[10] Kaelbling L.P., Littman M.L., Moore A.W. An Introduction to Reinforcement Learning. In: Steels L. (eds) The Biology and Technology of Intelligent Autonomous Agents. NATO ASI Series (Series F: Computer and Systems Sciences), vol 144. (1995) Springer, Berlin, Heidelberg.

[11] Y. LeCun, et al., Deep learning, Nature, 521 (2015), p. 436.

[12] A. Krizhevsky, et al.," Imagenet classification with deep convolutional neural networks Adv". Neural Inf. Process. Syst. (2012), pp. 1097-1105.
[13] H.F. Song, et al., "Reward-based training of recurrent neural networks for cognitive and value-based tasks
eLife" , 6 (2017), p21492.

[14] Bouneffouf, Djallel, Irina Rish, and Charu Aggarwal. "Survey on Applications of Multi-Armed and Contextual Bandits." *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020.
[15] Gupt, Ankit, Naveen Mysore Balasubramanya, and Mathini Sellathura. "Contextual-Bandit based MIMO Relay Selection Policy with Channel Uncertainty." *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020.
[16] Kohavi, Ron, and Roger Longbotham. "Online Controlled Experiments and A/B Testing." *Encyclopedia of machine learning and data mining* 7.8 (2017): 922-929.
[17] Goyal, Lakshay, et al. "Leaf and spike wheat disease detection & classification using an improved deep convolutional architecture." Informatics in Medicine Unlocked (2021): 100642.
[18] Deng, Wangdong, et al. "Thompson Sampling-Based Channel Selection Through Density Estimation Aided by Stochastic Geometry." *IEEE Access* 8 (2020): 14841-14850.
[19] D. Silver, et al." A general reinforcement learning algorithm that masters chess, shogi, and go through self-play Science" , 362 (2018), pp. 1140-1144.