A Survey on Multi-Armed Bandit with Deep Learning

YoungIn Kwon
younginkwon@unist.ac.kr
Ulsan National Institute of Science and Technology
Ulsan, South Korea

Abstract

Multi-Armed Bandit(MAB) is one of the sequential decision making algorithms by maximizing expected reward among multiple arms over the rounds. Bandit algorithms have utilized the optimism-in-the-face-of-uncertainty(OFU) principle what stressed to solve the exploration-exploitation trade-off. For estimating adequate arm of each round, there are statistical assumptions and techniques with previous information of decisions. By further research papers, there are main algorithms to decide optimal arm such as ϵ-greedy, UCB, Thompson Sampling, Actor-Critic, and SqaureCB. However, deep learning also has adopted at bandit algorithm as estimating optimal choice. In this paper, the advantages and disadvantages of main algorithms will be reviewed. And, same properties will be discussed in deep learning approach like other algorithms do. Lastly, the topic which promising deep learning techniques can be applied by bandit's mechanism will be considered.

Keywords: sequential decision making, multi-armed bandit, deep learning

1 Introduction

Bandit is one of the sequential decision making algorithm for finding optimal arm at specific time by previous taken rewards and pulled arms. It was introduced[14] and studied[9] as the standard stochastic d-armed bandit problem. Recently, it is called Multi-Armed Bandit(MAB) algorithm because there are several choices what need to be decided at one time. For solving bandit problem, the optimism in the face of uncertainty(OFU) approach has utilized. The objective of bandit is handling exploration-exploitation trade-off for finding optimal sequence of arms which maximize expected reward over the rounds as this approach.

However, in bandit problem, the notion named regret used for evaluating performance. It is the difference of expected reward between optimal sequence and bandit selected sequence. The equation getting regret could be different to depend on bandit environment. For instance, there is regret equation[10] as stochastic bandit which has collection of distribution $v = (P_a : a \in A)$ per each arms below:

$$R_n(\pi, v) = n\mu^*(v) - \mathbb{E}[\sum_{t=1}^{n} X_t] \tag{1}$$

Like definition of regret above, there are two parts in the right-hand side of equation. Left one is optimal reward followed by sequence of optimal strategy. Because $\mu^*(v)$ represents the largest mean as $\mu^*(v) = max_{a \in A}\mu_a(v)$ and $\mu_a(v)$ represent mean of action a as $\mu_a(v) = \int_{-\infty}^{\infty} xdP_a(x)$. Right part of right-hand side means expected reward of bandit sequence.

Even though there are differences of regret equation, there is common property which is always greater than 0. If bandit is more getting trained, regret would be decreased. However, the amount of regret keep increasing because bandit could do exploration despite many iteration done. Therefore, for measuring performance of bandit, the bound of regret needs to be proposed as presenting the level of maximum regret reached. The best regret bound revealed as sub-linear form.

There are several type of bandit called stochastic, adversarial, and contextual bandit. The first emerged bandit was stochastic bandit what optimal arms are determined by distribution of stacked previous trials. The probability of arms will be estimated optimal by selecting optimal arms per each round by algorithms utilizing previous taken action and gotten reward. Secondly, adversarial bandit was from pragmatic approach than stochastic bandit because, in real world problem, the arms could not have specific probability. Instead of the assumption, it has a policy π what maps previous information to action competing optimal action in hindsight. Even though it released strong assumption, it still had similar regret bound with stochastic bandit. Lastly, there is contextual bandit that is adopting context information of arms in decision making. The reason why utilizing context information is that bandit had not adopted additional context until it appeared. There are lots of context stuff in ordinary data such as demographic, preference and so on. It considered what if there is one action that has context information, the other arms that have similar context will be updated by context.

Behind the kinds of bandit in broad insight, there are main algorithms what have good performance with evolving movement of bandit. There are five algorithms will be discussed later. And there are ϵ-greedy[15], UCB[2, 4, 9], Thompson sampling[16], Actor-Critic[11], and SqaureCB[7]. ϵ-greedy is the method to make exploration at ϵ probability and exploitation $1-\epsilon$ at time round. Upper Confidence Bound(UCB) is the way choose reward's highest confidence bound contained expectation and standard deviation of reward among the arms. Thompson sampling is the solution adapting random sampling method at UCB algorithm. Actor-Critic is the approach to divide actor and critic as arm selector and reward

model. Lastly, SquareCB is the way to control exploration using difference of estimates between arms.

By those algorithms, bandit has influenced at some real-life application and supporting machine learning area[5]. There are few examples but recommender system is discussed little bit. In recommender system, there is typically suffered problem called cold-start problem when the amount of information is not enough to recommend due to lack of interaction. To overcome this problem, supplement information such as contexts what the user and item have used. However, bandit is considered as good solution to handle cold-start problem by adjusting strength of exploration and exploitation. And one of the main advantages to utilize bandit in recommender system is that it can online evaluation compared with conventional collaborative filtering method. As supporting machine learning, bandit can support some selection techniques for algorithm selection, hyperparameter tuning, and feature selection. For instance, in hyperparameter tuning, various hyperparameter combinations by the pre-defined range of parameter determined from user are the arms of bandit problem. The optimal parameter of model will be discovered by optimization through bandit mechanism.

The objective of bandit is estimating uncertainty. Recently, deep learning method have been adopted at estimation combined conventional UCB method such as neural UCB. Except of this paper, there are various combination between conventional algorithm and deep learning. In this paper, recently suggested deep learning or neural network methods in bandit are discussed. And the topic which other deep learning techniques can be applied at bandit will be discussed from proposed researches.

2 Related Works

2.1 Stochastic Bandit

In stochastic bandit setting, arms are estimated by reward distribution from information of previous rounds such as pulled arm and reward. That's why it's called stochastic compared with other bandits.

However, there were some trial and error[3, 12, 17] before the characteristic of bandit environment was figured out in general. It is that individual data of dataset is not i.i.d setting in bandit environment. In other words, the data in specific round is produced by pulled arms and reward of previous rounds. Due to this property, researchers who claimed no difference between them failed to prove that regret is bounded followed by bandit trials.

For dealing with this problem, new martingale technique[1] proposed as estimating coefficient at each round and it had better regret bound previous algorithms. In stochastic bandit setting, linear keyword occurred frequently because the estimation is faster and easier by restricting reward function as linear. The reward function is linear in this paper as well.

2.2 Adversarial Bandit

In real world, some player who pulls the arm as bandit could perform volatile way looked like random selection. So, it said that there is no specific distribution at the arms unlike stochastic setting. It's called adversarial bandit. Even though it just have one restriction what the reward is bounded, the regret is finally bounded similar with stochastic bandit.

Here, the arm will be chosen by adversarial cost per expert who decide next arm. The N number of experts calculates costs to all possible actions. By utilizing summation of all action's cost per expert, the expert what has minimum total cost will be selected from probability updated by bandit trials initialized as uniform distribution.

2.3 Contextual Bandit

Contextual bandit is utilizing side information such as demographic of user and property of action unlike conventional bandit[12]. By exploiting additional context, it helps to enhance prediction performance. It could be adopted at stochastic and adversarial bandit at the same time because it is not mutual exclusive between those. It is similar to acquire context vector at the beginning of each rounds like normal bandit problem with vector of arms. And it is used at pulling the arm by algorithms.

3 Main Algorithms

3.1 ϵ-greedy

At first, take a look at simplest policy called ϵ(epsilon)-greedy[15]. It's based on Explore-Then-Commit[10] method but there is randomised property which takes empirically best arm in $1-\epsilon$ and explore uniformly in ϵ in exploitation step unlike original method.

Explore-Then-Commit method is the way dividing preliminary m exploration step and $n-m$ exploitation step afterwards.

Algorithm 1: Explore-Then-Commit

input : m

In round t choose action

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

In evaluating regret bound process, It could be noticed that there is exploration/exploitation dilemma by the amount of m step within total n step. In regret inequality, left means exploration and right means exploitation. The those amounts will be differed by intensity of m.

$$R_n \leq m\sum_{i=1}^{k}\Delta_i + (n-mk)\sum_{i=1}^{k}\Delta_i exp(-\frac{m\Delta_i^2}{4}) \qquad (2)$$

For overcoming dilemma, ϵ-greedy is interleaving two process by arbitrary real value ϵ_t at each time t without preliminary exploration.

Algorithm 2: ϵ(epsilon)-greedy

input: k, T, $\epsilon \in \{0, 1\}$
initialize for a=1 to k:
$\hat{\mu}_a(0) \leftarrow 0$
for $t \in 1, ..., T$ **do**

$$A_t \leftarrow \begin{cases} argmax_i\ \hat{\mu}_i(t-1) & 1-\epsilon \\ \text{random uniform action} & \text{with probability } \epsilon \end{cases}$$

$$X_t \leftarrow bandit(A_t)$$

$$T_i(t) \leftarrow \sum_{s=1}^{t} \mathbb{I}\{A_s = i\}$$

$$\hat{\mu}_i(t) \leftarrow \frac{1}{T_i(t)} \sum_{s=1}^{t} \mathbb{I}\{A_s = i\} X_s$$

end

3.2 UCB

Secondly, there is Upper Confidence Bound(UCB) Algorithm[2, 4, 9] as bandit algorithm. In this perspective, it based on the principle of optimism in the face of uncertainty. The uncertainty is confidence interval composed by empirical mean and exploration intensity. Through this interval, bandit get optimistic largest value it could plausibly be. Determining the degree of uncertainty(δ) is important because it controls width of confidence interval in this algorithm. It will influence the result which action is pulled.

$$UCB_i(t-1, \delta) = \begin{cases} \infty & \text{if } T_i(t-1) = 0 \\ \hat{\mu}_i(t-1) + \sqrt{\frac{2log(1/\delta)}{T_i(t-1)}} & \text{otherwise} \end{cases} \quad (3)$$

Algorithm 3: UCB(Upper Confidence Bound)

input: k, T, δ
for $t \in 1, ..., T$ **do**
  Choose action $A_t = argmax_i UCB_i(t-1, \delta)$
  Observe reward X_t *and update upper confidence bounds*
end

There were lots of researches about UCB-based bandit algorithm. Among them, the LinUCB[6] will be introduced because it adopted the linear combination concept in estimation of bandit, overcame computational complexity compared with previous work, and utilized web advertisement system practically as well.

3.3 Thompson Sampling

The third algorithm is Thompson Sampling[16] which is called Posterior Sampling as well. That's because that it samples action by posterior distribution which is configured from pre-defined prior distribution. Then, the posterior distribution keep updating based on observation of sampled action. At the beginning of this algorithm, it was utilized at simple bandit case such as bernoulli bandit which has 2 choices per actions. It recently has used at broad bandit environment with theoretical guarantees. Specifically, it mainly appears in finite-armed stochastic bandit and linear stochastic bandit environment.

Algorithm 4: General Thompson Sampling($\mathcal{X}, p, q, r$)

procedure *THOMPSONSAMPLING*($\mathcal{X}, p, q, r$)
  for *t=1,...,T* **do**
    Sample model.
    Sample $\hat{\theta} \sim p$
    Select and apply action:
    $x_t \leftarrow argmax_{x \in \mathcal{X}} \mathbb{E}_{q\hat{\theta}}[\mathrm{r}(\mathrm{y}_t)|x_t = x]$
    Apply x_t and observe y_t
    Update distribution:
    $p \leftarrow \mathbb{P}_{p,q}(\Theta \in \ . \ |x_t, y_t)$
  end
end procedure

Unlike normal stochastic bandit problem, it has one more element as prior distribution like $(\mathcal{X}, p, q, r)$, outcome space, prior distribution, conditional probability measure, and reward function. The difference is taking prior distribution of every arm before sampling. It will keep updating by sequence which samples , selects action by maximizing expected reward, observes reward from chosen action.

Thompson sample would better than ϵ-greedy. The reason is that there is bias in sampling process. The exploration is affected by concentration of posterior distribution. Posterior distribution will be reflected well to unknown true $theta$ by increasing number of iteration. It means it will explore possible good action. However, ϵ-greedy pull the arm as uniform distribution. And, if we compare with UCB algorithm handled above, thompson sampling doesn't need the function to get upper confidence bound. But, it needs to restriction at reward distribution such as sub-gaussian distribution unlike UCB.

3.4 Further Algorithms

After three basic bandit algorithms, there are many trials to figure out another methodologies. Here, two state-of-the-art methodologies are introduced as Actor-Critic and sqaureCB.

Actor-Critic[11] for bandit proposed as first time. This method also has same objective to figure out optimal policy in linear stochastic bandit setting. Actor estimates parameter of reward function based on previously chosen action

and reward and Critic estimates expected rewards by estimated parameter of reward function. Algorithm proceeds the sequence of Critic and Actor from initialized θ_0 parameter.

Next, SqaureCB[7] was proposed as reduction contextual bandit problem to online regression oracle. In simple speaking, it is mapping contextual bandit problem to regression problem by squared loss function. So it called SquareCB.

4 Deep Learning approach

In this section, deep learning adopted approach will be discussed with background of needs to use it in bandit at first. And then, two profitable methodologies are introduced as subsection of this chapter. Lastly, the beyond perspective with prior researches using deep learning will be proposed.

In stochastic bandit environment, the distribution of reward function should be restricted by definition. So, the objective of stochastic bandit is estimating parameter of unknown reward function. The majority of researches restricts the function as linear due to online learning characteristic of bandit, even if function could seem like non-linear in real world problem. Those are the reason why deep learning approaches are adopted in bandit problem. Neural network has enormous power to approximate random function and can overcome non-linear problem by stacking multiple layer.

However, there are some problems when applying it. There might be computational and delay issue at estimating optimal arm due to lots of learning parameter of network. It will lose online advantage of bandit. And the assumption which reward function is non-linear is not always adequate in all data like one research saying introducing later subsection.

4.1 Neural UCB

The idea of Neural UCB[18] is aligned with the background to estimate parameter of unknown reward function by neural network. Then, it constructs upper confidence bound for exploration. Neural network utilizes for getting expected mean of UCB which contained mean and exploration intensity. This neural net is updated by TrainNN algorithm which has own loss function.

Algorithm 5: Neural UCB

intialization :Randomly Initialize θ_0 as described in the paper
initialize $Z_0 = \lambda \mathbf{I}_{d \times d}$
for t=1,...,T **do**
  Observe context vector for all K arms
  for a=1,...,K **do**
    Compute upper confidence bound
    Sample greedy action among bounds
  end
  Play sampled action and observe reward
  Update Z_t by gradient between observation and estimator
  Update θ_t by **TrainNN**
  Compute scaling factor γ_t
end

4.2 NeuralLinear

NeuralLinear[13] was appeared as combined neural network of thompson sampling. Thompson sampling is estimating posterior probability of reward function for estimating optimal policy. If reward function assumed as linear function, the posterior function by thompson sampling is also linear. The representation power would be problem when the reward function is defined as linear. To overcome this, it is to utilize neural network for estimation which is non-linear representation of last layer of neural network.

In this method, neural network and posterior estimation could be considered respectively because just input variables are different compared with normal thompson sampling. So, it can train neural network and regression at the same time. It has two advantages like above, however, the neural network could not be updated per times when data gathers enough for training.

4.3 Future Directions

According to two bandit implementations using deep learning, the main advantage is powerful capability of function approximation. But, it spends more time than methodology without deep learning. So, the final objective of bandit with neural network is consuming little time in function approximation as well as not losing representation power.

As next neural architecture of bandit, 1D CNN technique which has feature extraction and classification process at one dimensional time-related signal could be applied because it have similar data property with bandit, computational efficiency, and representation capability. The significance of 1D CNN investigated by practically and theoretically as state-of-the-art solution in time-series signal data[8].

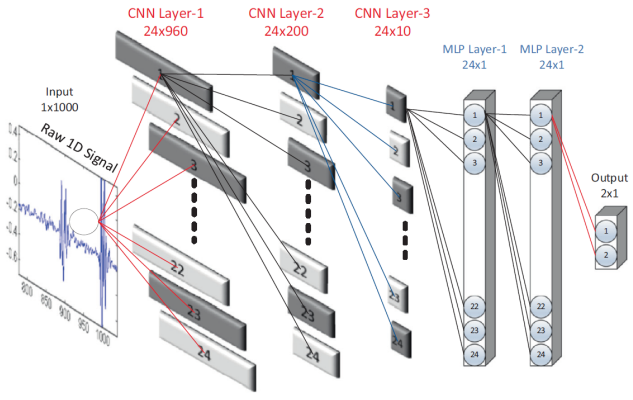

Figure 1. A sample 1D CNN (3-CNN and 2-MLP layers)[8]

The overall shape of 1D CNN is similar with image based 2D CNN. However, kernel defines and stride one dimensional space unlike normal CNN. So, it strides row-wise direction per the amount of kernel size. Through 1D CNN, it learns time-related representation from time-based signal data. And,

there are advantages of 1D CNN instead of 2D CNN in low computational complexity, decreased number of parameters, and adopting real-time and low-cost application.

From the characteristic and advantages of 1D CNN model, it could be plugged as bandit problem whether last MLP layer considered as last hidden layer of neural network based bandit. There are two reasons why it can apply as bandit algorithm. The first reason is that bandit data also has sequential characteristic like 1D CNN data. Secondly, the architecture will be proper as deep learning based bandit due to capability of feature representation and handling real-time application.

5 Conclusion

Bandit is algorithm which estimates parameter by maximizing expected reward to interact unknown environment along n rounds. The main concern of bandit problem is exploration-exploitation dilemma in estimate unknown parameter. In procedure figuring out this problem, the fact that bandit problem has different property as non i.i.d data unlike supervised learning discovered. And, bandit algorithms have improved based on conventional and modern approaches. In this paper, those things about overall background of bandit dealt with bandit definition, objective, environment, and algorithms.

Among them, deep learning based bandit algorithms have stressed due to emerged representation power of neural network as estimating parameter. There are two examples called Neural UCB and NeuralLinear from existing UCB and Thompson Sampling method. For the next step of those researches, bandit combined 1D CNN proposed at estimating parameter due to same property of data, small number of parameter, and representation power through feature extraction of CNN. However, it is ideal suggestion to cover drawbacks when neural network used in bandit and enhance representation power than multiple fully-connected layers.

In conclusion, the objectives of recent bandit researches are generalizing non-linear reward function and enhancing performance within preserve online learning characteristic.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits.. In *NIPS*, Vol. 11. 2312–2320.

[2] Rajeev Agrawal. 1995. Sample mean based index policies with O (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability* (1995), 1054–1078.

[3] Peter Auer. 2000. Using upper confidence bounds for online learning. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE, 270–279.

[4] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[5] Djallel Bouneffouf and Irina Rish. 2019. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040* (2019).

[6] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 208–214.

[7] Dylan Foster and Alexander Rakhlin. 2020. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*. PMLR, 3199–3210.

[8] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 2021. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing* 151 (2021), 107398.

[9] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.

[10] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[11] Huitian Lei, Ambuj Tewari, and Susan A Murphy. 2017. An actor-critic contextual bandit algorithm for personalized mobile health interventions. *arXiv preprint arXiv:1706.09090* (2017).

[12] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

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

[14] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.

[15] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

[16] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.

[17] Thomas J Walsh, István Szita, Carlos Diuk, and Michael L Littman. 2012. Exploring compact reinforcement-learning representations with linear regression. *arXiv preprint arXiv:1205.2606* (2012).

[18] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural contextual bandits with UCB-based exploration. In *International Conference on Machine Learning*. PMLR, 11492–11502.