Estimating Click Through Rate Using Reinforcement Learning Algorithms

[1]A.Lakshmanarao,[2]G.Vijay Kumar,[3]K.Chandra Sekhar
[1]Associate Professor,[2]Assistant Professor,[3]Assistant Professor
[1,2,3]Department of CSE
[1,2,3]Pragati Engineering College, Surampalem,AP,India
[1]laxman1216@gmail.com,[2]vijay9908914010@gmail.com,[3]chandrasekhar.k@pragati.ac.in

Abstract: Click-through rate (CTR) prediction is one of the important tasks in business applications like online advertising. It is an integral part of online advertising systems. It is used as an input to auctions which find the final ranking of ads. Machine Learning algorithms are very popular to solve interaction problems. Especially Reinforcement Learning algorithms work very well with these kinds of problems. In this paper, we proposed a model based on two algorithms Upper Confidence Bound and Thompson sampling for better click through rate prediction. The dataset is taken from superdatascience which consists of information about 10 ads. If a user click on the ad, reward is 1, otherwise it is 0.We applied reinforcement learning algorithms to this dataset and computed the total rewards for each of the algorithm separately.

***Index Terms*–Click-Through Rate (CTR), Machine Learning, Reinforcement Learning, Python**

I. INTRODUCTION:

Advertising through search engine has become a significant thing while coming across web browsing [1].These advertising generally using the keyword auction model, and they pay cost of keywords. Pay per click with a cost per click billing is used.
Click through Prediction is rigor sly studied in recent years.
Agarwal et al. [2] deals with two stages of click through prediction process. In first step, keywords are predefined and in second step, click through rate for each area is computed. Chakrabarti et al. [3] uses logistic regression and employed a multiplicative factorization to model the interaction effects for several regions.Regelson and Fain [4] proposed Click Through Rate using machine learning hierarchical clusters for the low frequency or completely novel terms. Zhipeng Fang, Kun Yue et al.[1] proposed Click-Through Rates of New Advertisements Based on the Bayesian Network.

The role of machine learning plays a significant factor in online advertisement serving. V. Chaoji et al.[5] explains how important the machine learning is in the real world.Muhammad Junaid et al.[6] proposed Click Through Rate Prediction for Contextual Advertisements Using Linear Regression.

Reinforcement Learning is one type of Machine Learning.It is also called as Online Learning. Reinforcement Learning algorithms are completely different from the supervised and unsupervised machine learning algorithms. Reinforcement Learning is generally used to solve interaction problems means that the data observed up to time k is used to take aaction at time k + 1.Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps; for example, maximize the points won in a game over many moves. They can start from a blank slate, and under the right conditions, they achieve superhuman performance. Like a child incentivized by spankings and candy, these algorithms are penalized when they make the wrong decisions and rewarded when they make the right one this is reinforcement. Reinforcement learning algorithms try to find the best ways to earn the greatest reward. Rewards can be winning a game, earning more money or beating other opponents. They present state-of-art results on very human task, for instance. There are three basic concepts in reinforcement learning: state, action, and reward.Dave and Varma [7] used the gradient boosting decision tree (GBDT) to predict the advertising CTR. They extracted similar features from advertising data and discovered implicit relationships between different features. Finally, they found out the nonlinear relationships between the predicted target and features. He et al. [8] introduced a fusion model which combines decision trees with logistic regression for predicting clicks on Facebook ads.

II.RELATED WORK:

The dataset contains 10 ads information in the form of rewards. All these ten are the different versions of the same ad.Here, we need to find which ad is best and suited to put on the social network.We will put the ad that has maximum clicks and best conversion rate. So here the task is find which version of this ad is the best for the user. This dataset is only the simulation. In real life, we are going to start experimenting and placing different version of ad on social network and based on the observed results, we will change our strategy to place these ads on social networks.
Each time a user of social network log in to account, we will place one version of these 10 ads and observe the response of user. If user clicks on the ad, reward 1 is added, otherwise reward is 0.This task is repeated for 10000 users.

Random selection algorithm:

This algorithm works by selecting random version of the ad at each round. The sum of the rewards up to last round is calculated. A random range function is used.

Sample code:

```
N = 10000
d = 10
ads_selected = []
total_reward = 0
for n in range(0, N):
    ad = random.randrange(d)
    ads_selected.append(ad)
    reward = dataset.values[n, ad]
    total_reward = total_reward + reward
```

After applying this algorithm, we get a reward of 1203.In second run, we get a reward of 1192.Since we applied a random function, we may get a different result each time. But the values are nearer to each other.

This Random algorithm is compared with the machine learning algorithms Upper Confidence Bound algorithm, Thompson Sampling algorithm.

III.PROPOSED MODEL:

In the proposed system, a sample dataset is taken from superdatascience.Afterthat, data reprocessing step is applied to imported dataset.In data pre-processing step, verify if dataset contains any missing values.If it contains try to replace missing values with mean or median. Data preprocessing also applies Feature scaling, label encoding depending on dataset.Then,random selection algorithm is applied.Afterthat, we applied Upper Confidence Bound algorithm and see that any improvement over random selection algorithm. And finally, we applied Thompson Sampling reinforcement algorithms on dataset and compared results. Python Programming language was used for implementation purpose.

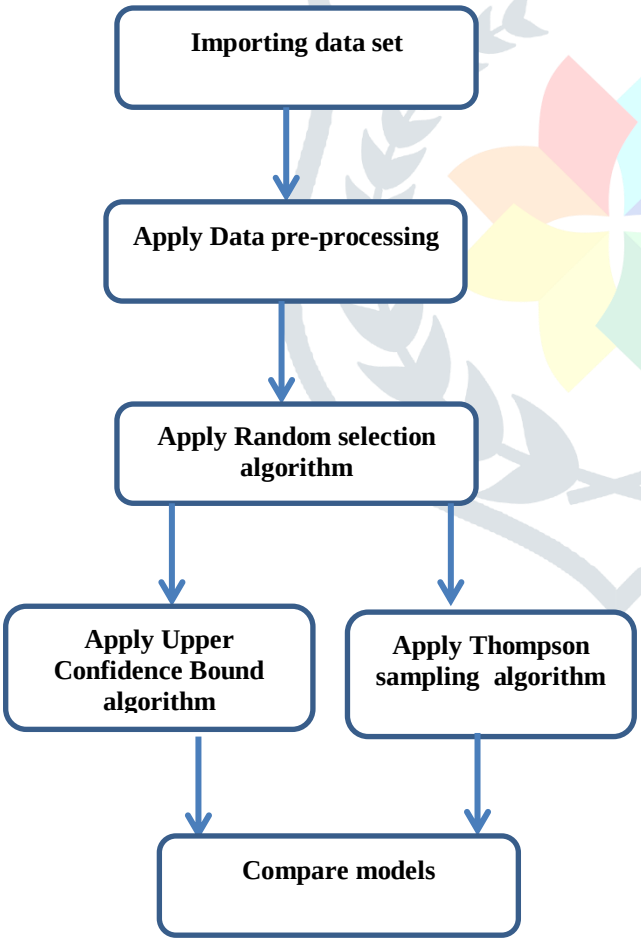

Figure 3.1: Proposed model

IV.EXPERIMENTATION & RESULTS:

In our experiments, weopted pythonlanguage. Because Python language supports several packages for implementing machine learning techniques. The dataset used for this work is taken from UCI Repository.In this dataset, there are 9568 instances of five features namely Temperature(T),Exhaust Vacuum(V),Ambient Pressure(AP),Relative Humidity(RH),net hourly electrical energy output (EP) of the plant.in this the feature EP is taken as dependent variable and remaining four are independent variables.We need to predict the net hourly energy output based on four parameters.

Upper Confidence Bound:

Step1:
At each round n, we consider two numbers for each ad i.
Ni(n) is the number of times the ad I was selected up to round n
Ri(n) is the sum of rewards of the ad I up to round n.
Step 2:
From these two numbers we compute:
- The average reward of ad i up to round n

ravg(n)=Ri(n)/Ni(n)
the confidence interval [ravg(n)-Δi(n),ravg(n)+ Δi(n)] at round n with
Δi(n)=sqrt(3log(n)/2Ni(n))
Step3:
We select the ad I that has the maximum UCB ravg(n) + Δi(n)

Sample code:

```
N = 10000
d = 10
ads_selected = []
nos_of_selections = [0] * d
sumofrewards = [0] * d    #basically step1 done
totalreward = 0
for n in range(0, N): # 0 to 10000 for each round
    ad = 0
    maximum_upperbound = 0 # it is diff at each round, so initialize at each round
    for i in range (0, d): #for each version of add
if (nos_of_selections [i] > 0):#for round 0(we dont have much information) if add ver i selected at least once then use this strategy
            avgreward = sumofrewards [i] / nos_of_selections [i]
            delta_i = math.sqrt(3/2 * math.log(n + 1) / nos_of_selections [i])
            upperbound = avgreward + delta_i  # step 2 done
        else:
            upperbound = 1e400 #verylarge value (10power 400)
        if upperbound > max_upperbound:
            max_upperbound = upperbound
            ad = i #track index with max upper bound each time
    ads_selected.append(ad)
numbers_of_selections[ad] = numbers_of_selections[ad] + 1
    reward = dataset.values[n, ad]
    sums_of_rewards[ad] = sums_of_rewards[ad] + reward
    total_reward = total_reward + reward
```

Histogram of ads:

From the histogram, we observed that ad 4 has more click through rate. So the best ad for this dataset is 4 because it has highest conversion rate.

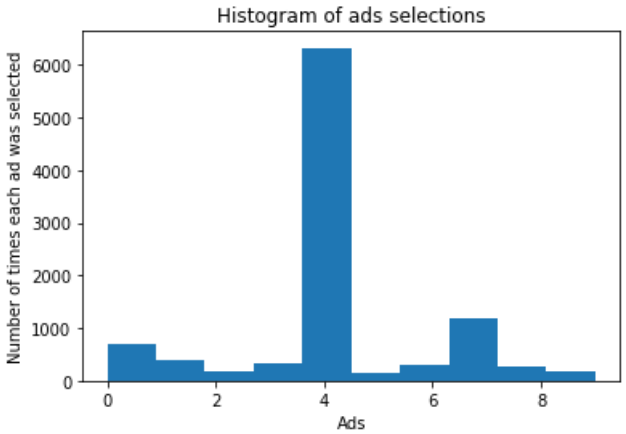

Figure 4.1:Upper Confidence Bound

Thompson Sampling:

Step 1:
At each round n, we consider two numbers for each ad i:
Ni1(n) is the number of times the ad I got reward 1 up to round n.
Ni0(n) is the number of times the ad I got reward 0 up to round n

Step 2:
For each ad i, we take a random draw from the distribution $\delta i(n) = \beta(Ni1(n)+1, Ni0(n)+1)$

Step 3:
We select the ad that has the highest $\delta i(n)$

Sample code:

```
N = 10000
d = 10
ads_selected = []
nosof_rewards_1 = [0] * d
nosof_rewards_0 = [0] * d
totalreward = 0
for n in range(0, N):
    ad = 0
    maxrandom = 0
    for i in range(0, d):
        randombeta = random.betavariate(nosof_rewards_1[i] + 1, nosof_rewards_0[i] + 1)
        if randombeta > maxrandom:
            maxrandom = randombeta
            ad = i
    ads_selected.append(ad)
    reward = dataset.values[n, ad]
    if reward == 1:
        nosof_rewards_1[ad] =    nosof_rewards_1[ad] + 1
    else:
            nosof_rewards_0[ad] =       nosof_rewards_0[ad] + 1
    totalreward = totalreward + reward
```

After applying this algorithm,we get a reward of 2597.In second run we get 2452.This is not a fixed value,but compared to Upper Confidence Bound ,Thompson Sampling gives better result.

Histogram of ads:

From the histogram, it is observed that ad 4 is the best ad; because it has better conversion rate. We got same result in Upper Confidence Bound also.

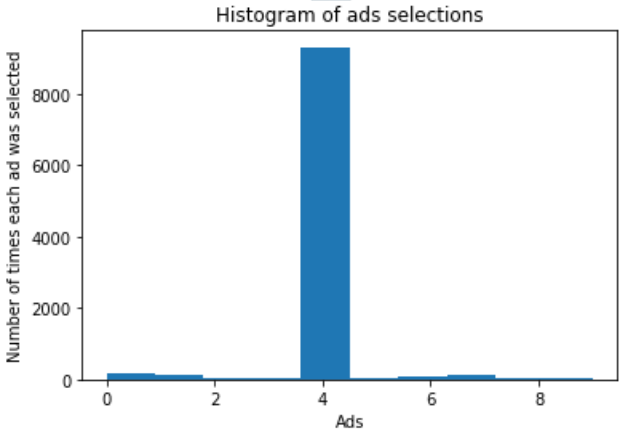

Figure 4.2:Thompson Sampling

Comparison Of Results:

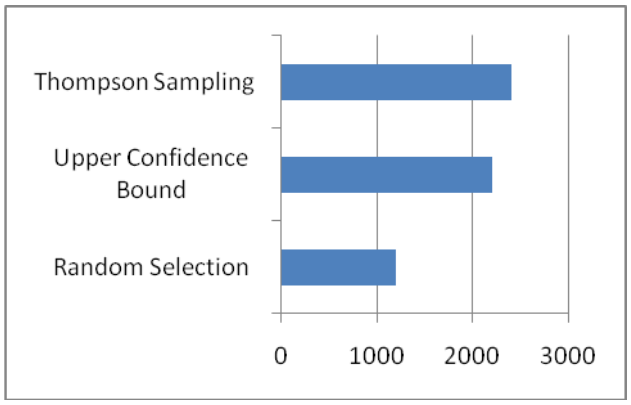

Figure 4.3:comparision of algorithms

IV.CONCLUSION:

Click through Rate prediction in online advertisement was analysed.First a random selection algorithm is applied.For the proposed dataset, this algorithm gives a reward of 1200(approximately).After that Upper confidence bound algorithms was applied.A reward of 2200 was obtained.Later Thompson Sampling algorithm was applied.A reward of around 2400 was obtained.Reinforcement algorithms given better performance over random algorithm, especially Thompson sampling algorithm given better performance for Click Through Rate Prediction.

REFERENCES:

[1] Zhipeng Fang, Kun Yue Jixian Zhang, Dehai Zhang, and Weiyi Liu "Predicting Click-Through Rates of New Advertisements Based on the Bayesian Network ",Hinndawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 818203.

[2]D. Agarwal, A. Z. Broder, D. Chakrabarti, D. Diklic, V. Josifovski, and M. Sayyadian, "Estimating rates of rare events at multiple resolutions," in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07), pp. 16–25, August 2007

[3] D. Chakrabarti, D. Agarwal, and V. Josifovski, "Contextual advertising by combining relevance with click feedback," in Proceeding of the 17th International Conference on World Wide Web (WWW '08), pp. 417–426, New York, NY, USA, April 2008.

[4]M. Regelson and D. Fain, "Predicting click-through rate using keyword clusters," in Proceedings of the 2nd Workshop on Sponsored Search Auctions, 2006

[5]V. Chaoji, R. Rastogi, and G. Roy, "Machine Learning in the Real World," Proc. VLDB Endow., Vol. 9, No. 13, 2016.

[6]Muhammad Junaid Effendi, Syed Abbas Ali, "Click Through Rate Prediction for Contextual Advertisement Using Linear Regression", International Journal of Computer Science and Information Security, • December 2016.

[7]Dave K., Varma V. Predicting the Click-through Rate for Rare/New Ads. Hyderabad, India: Center for Search and Information Extraction Lab International Institute of Information Technology; 2010.

[8] Bowers S., Candela J. Q., et al. Practical lessons from predicting clicks on ads at Facebook. Proceedings of Eighth International Workshop on Data Mining for Online Advertising; August 2014; New York, NY, USA.