NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science and Business Analytics"

UDC 004.8

Research Project Report on the Topic:
Contextual Bandits for Recommendation Systems

Fulfilled by:

Student of the group #БПАД212
Gorelskii Roman Evegenievich

(signature) (date)

Assessed by the Project Supervisor:

Samsonov Sergey Vladimirovich
Research Fellow
Faculty of Computer Science, HSE University

Moscow 2024

Contents

Abstract

The implementation of recommender systems' algorithms remains to be one of the most crucial parts for many companies. It allows them to make the user experience to be more personalised. Finding a solution which adapts to the dynamic change in the pools of content and the scale of web services expansion is what modern developers are trying to implement. This research is aimed at the exploration and evaluation of the contextual bandit approach. It allows to reduce the computational complexity with the possibility to test the performance on the traffic recorded previously.

Аннотация

Внедрение алгоритмов рекомендательных систем остается одной из важнейших задач для многих компаний. Это позволяет им сделать взаимодействие с пользователями более персонализированным. Современные разработчики пытаются найти решение, которое адаптировалось бы к динамичным изменениям в наборах контента и масштабам расширения веб-сервисов. Данное исследование направлено на изучение и оценку подхода контекстуальных бандитов в рекомендательных системах. Их использование позволяет снизить вычислительную сложность с возможностью тестирования качества на собранном ранее трафике.

Keywords

Big data, Contextual bandit, Machine learning, Deep learning, Recommender system

1 Introduction

1.1 Subject Area

Recommender system is a machine learning based algorithm which is aimed at the analysis of big data which helps to provide users with more relevant products for them. It is implemented in various spheres ranging from personalised advertisements to playlist creations and movie suggestions on various platforms. The problems which these systems face are the growing size of network services and increasing speed of content variation.

One of the most widely used formulation is in terms of multi-armed bandit (MAB) problem. The algorithm's strategy is to select the most optimal variant (arm) for recommendation.

1.2 Problem Statement

1.2.1 Exploration vs Exploitation

The main dilemma of reinforcement learning in terms of MAB is the trade-off which appears during the selection of the best strategy. Formulating it in terms of multi-armed bandit terminology, the decision consists in whether to explore the effect from the choice of some arm or to exploit the arm which brings greatest reward at the current time step.

1.2.2 Mathematical Statement of the Problem

The stated description allows to formulate the efficiency criteria which is ought to be used for the definition of the mathematical statement. Let a_t describe some chosen arm and a_{t^*} be the selected arm according to the optimal policy at a time step t. Then r_{t,a_t} describes the reward (feedback) received from the choice of a_t ($r_{t,a_{t^*}} >= r_{t,a_t}$). In this case the regret value to be minimised is defined as following:

$$R_A(T) = E[\sum_{t=1}^{T} r_{t,a_{t^*}}] - E[\sum_{t=1}^{T} r_{t,a_t}]. \tag{1}$$

Such a formulation can be applied to any bandit algorithm. However, the optimal policy is usually unknown for the algorithm due to the complex structure of dynamic pools of data. Hence, to perform the comparison of different algorithms the efficiency criterion can be reformulated in terms of click-through rate (CTR). This metric is often used because of its simplicity in calculation. It's especially convenient in the case of a binary reward where the observation of "1" as feedback can be interpreted as a user click ("0" ≡ "absence of click"). It can be formally defined as:

$$CTR_t = \frac{CumRew_t}{t}. \tag{2}$$

where term $CumRew_t$ equals to the $\sum_{i=1}^{t} r_{i,a_i}$. As a result, CTR is maximised for every t.

1.3 Project Structure

The work is organised in the following way. Section (2) presents the overview of the literature used during the project. Afterwards, Section (3) provides with the description of data, CTR simulation and results obtained from the implementation of $\epsilon - greedy$ and LinUCB algorithms. Section (4) expands the research further with the exploration of possible incorporation of Thompson sampling ideas for contextual bandits. The final thoughts and proposals are presented in the Section (5).

2 Literature Review

2.1 The Task of Analysing the Previous Methods of Implementation

One of the most basic algorithms used to solve the multi-armed bandit problem is called $\epsilon - greedy$. Introduced in [3] such an implementation aims at providing the policy which allows to balance between the exploration and exploitation. It is done through the introduction of ϵ likelihood variable which is used to randomly decide whether to select an arm by chance or to use the best current one according to some history at the current time step (H_t).

This algorithm has several reasons to be selected as a benchmark. Firstly, its simplicity in comprehension and implementation allows to explain the underlying basic ideas of the problem. Secondly, this unguided algorithm emphasizes the improvements that are brought by other algorithms regarded in this paper.

2.2 Basic Principles of LinUCB Algorithm

The research itself is mostly based on the findings of [4]. It is not a unique example of an algorithm that assumes linear expected payoffs [2]. However, its intuition is more accessible for comprehension of the context impact on recommendations. The main idea of the authors was to provide an algorithm that computes the upper confidence bound, used for the efficient balance between the exploration and exploitation processes in the contextual setting of bandit problem. It is stated that there exist two types of linear models: disjoint and hybrid. The first one differs from the second in the aspect that it only uses the parameters not shared among other arms. Nonetheless, the mathematical description of optimal arm selection of the disjoint model is similar to the hybrid case and can be described as:

$$a_t = \arg\max_{\alpha \in \mathcal{A}_t}(x_{t,a}^T \hat{\theta}_a + \alpha \sqrt{x_{t,a}^T (D_a^T D_a + I_d)^{-1} x_{t,a}}), \tag{3}$$

where $x_{t,a}^T$ is the context; $\hat{\theta}_a$ is an estimated coefficient; $x_{t,a}^T \hat{\theta}_a$ is the expected payoff of an arm a; $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$ for any $\delta > 0$; D_a is the design matrix; I_d is the identity matrix.

Providing the results of the experiments which the authors conducted based on the click-through rate of the front page of “Yahoo!”, it was concluded that the newly proposed method not only outperforms the previous ones but also is more effective for sparse data.

2.3 Incorporation of Thompson Sampling Ideas

Further step of the project is to explore the enhancement of the implemented LinUCB algorithm with the ideas of Thompson sampling. It is a popular approach used for multi-armed bandit problems which introduces the probability distribution for each expected payoff. This helps to perform the most optimal arm selection which minimizes the total regret.

One of the proposals presented in [1] suggests to assign a prior distribution for each arm. In this way the reward can be approximated with some parameters. The precision of recommendation is improved with the calculation of the posterior. Such an approach allows to incorporate the role of context in a different way compared to [4] while still remaining linearity among payoffs.

The idea presented in [5] pushes the research further and introduces the non-linearity into the estimation of rewards. The model incorporates the sequential neural network which helps to approximate the distribution of each arm. Such a proposal potentially leads to even more stable and efficient results compared with previous algorithms.

3 Description of Applied Methods and Results

3.1 Data Preprocessing

For the performance of simulations with various bandit algorithms the MovieLens dataset was chosen. It is one of the most popular options chosen by many researchers. It contains information about movies and ratings left by users (downloaded with Polara framework) as well as the data about them (downloaded with Pandas). In the setting of MAB problem the action of film recommendation can be viewed as a pulled arm.

userid	movieid	rating	timestamp
1	1193	5	978300760
1	661	3	978302109
1	914	3	978301968

userid	gender	age	occupation	zipcode
1	F	1	10	48067
1	M	56	16	70072
1	M	25	15	55117

movieid	moviename	genres
1	Toy Story (1995)	Animation|Children's|Comedy
2	Jumanji (1995)	Adventure|Children's|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance

Table 3.1: First 3 rows of "ratings", "users", "movies" dataframes

Some preprocessing steps were conducted with every dataset. In "users" dataframe one hot encoding was used to transform "gender" feature. Also, the each of 21 occupations were separated into individual columns and encoded for every person. Same procedure was done for the "genres" feature of "movies" dataset.

The most important preprocessing steps were performed with "ratings" dataframe. Firstly, to both facilitate the computational difficulty and receive relevant results during the training procedure only 30 most often rated films were chosen. Moreover, in the current research the binary split of reward was selected in a way that only movies that received rating 5 from the user contained "1" in the correspondingly named column ("0" if "rating" is less than or equal to 4). Lastly, the rows (observations) were randomly shuffled in order to simulate the online learning environment.

The performed steps allowed to prepare the information for correct training of implemented algorithms. Most importantly the nature of binary reward allows to use the CTR metric for comparison.

3.2 $\epsilon - greedy$ Implementation

Following the description presented in Section (2.1), the algorithmic representation can be formulated as following:

```
Algorithm 1 $\epsilon - greedy$ policy
Require: $H_t$                                  ▷ Collected history at the time step $t$
  $\epsilon \leftarrow$ random()                ▷ Likelihood to explore
  if $\epsilon = 1$ then
      Draw random arm
  else
      Draw the arm bringing highest reward w.r.t. $H_t$
  end if
```

Besides the definition of ϵ given in the comment, it can also be viewed as the proportion of time assigned for the algorithm to spend for exploration.

3.3 LinUCB Implementation

The usage of contextual bandits is aimed at proposition of more effective ways to balance the outcomes from the trade-off. This approach allows to use the context about the users in order to emphasize the usage of particular arms which potentially bring higher reward. The LinUCB approach is one of the possible ways to achieve it.

Its discussion must begin with the explanation of how the upper confidence bound is defined. The intuitive interpretation as that it is a form of a confidence interval for the expected payoff which allows to incorporate the level of uncertainty about the obtained reward.

From the mathematical point of view the decision of pulling an arm (recommending a film) is based on whether it maximizes the UCB (described by the Equation (3)). The term $D_a^T D_a + I_d = A_a$ is obtained from the decomposition of $\hat{\theta}_a$ with ridge regression. As a result, $\sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}$ is the estimated standard deviation that we want to incorporate. The parameter α regulates what is the level of uncertainty to be presented in the UCB. Formulating it in terms of trade-off dilemma, how much emphasis the exploration is provided with.

The algorithm has two variants of implementations: disjoint and hybrid. The difference between them is that the second one takes into account not only the similarity between the users but also between the selected arms. In the scope of the research movies can have several genres. So, recommending movies of particular kind to similar users might bring higher reward. This assumption can be reflected mathematically through the introduction of features ($z_{t,a}$) in the expected payoff: $E[r_{t,a}|x_{t,a}] = z_{t,a}^T \beta^* + x_{t,a}^T \theta_a^*$. The calculation of standard deviation also incorporates the influence of arms' similarities.

3.4 Click-through Rate Simulations Realisation

The empirical study and comparison of algorithms' performance is done with the help of CTR (described at Section 1.2.2). In order to simulate the environment of user's click the following version of the algorithm (referenced from bandit_simulations and adapted for the current study) is proposed:

Algorithm 2 CTR simulations

```
Require: Number of epochs ($E$)
  time_steps ← 0
  $CumRew \leftarrow 0$
  $CTR \leftarrow []$
  for $t$ from 0 to $E$ do
      Obtain information about the selected movie and reward
      Draw an arm ($a_t$) according to algorithms policy
      if $a_t \equiv$ "movie selected by user" then
          Update information for the bandit
          time_steps += 1
          $CumRew$ += reward
          $CTR \leftarrow CumRew/time\_steps$
      end if
```

There are several modifications which are introduced in the simulation with different bandits. Considering $\epsilon - greedy$, the updated information (expansion of known history H_t) happens when the algorithm recommends the coinciding film. Meanwhile, in the case of LinUCB bandit the selected arm's design matrices are updated with incoming context.

In the focus of the paper the simulations for every type of algorithm were conducted for 3 epochs in order to receive sufficient amount of training observations.

3.5 Click-through Rate Simulations Results

Applying the simulation function described at Section (3.4), the following results were obtained for $\epsilon - greedy$ algorithm:

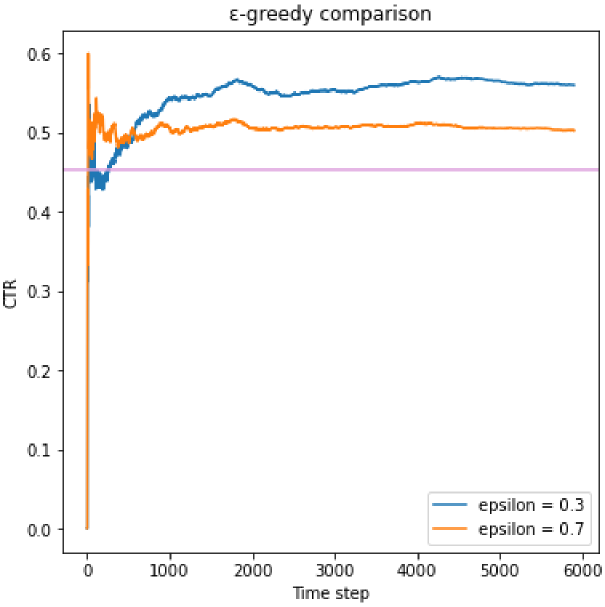

Figure 3.1: Comparison of $\epsilon - greedy$ algorithms

The plum coloured line depicts the average reward which can be obtained using random recommendations. It is plotted in order to highlight the correctness of the coded algorithm. This is verified by the CTR reaching $\approx 55\%$ for $\epsilon - greedy$ with emphasis on exploitation and $\approx 50\%$ with the one on exploration.

The Figure (3.1) helps to emphasizes the role of ϵ in the balancing of the trade-off. When it is equal to 0.3 the bandit spends more time on the exploitation of the best variant according to the collected history. When the value of *epsilon* is increased to 0.7, the emphasis is stronger on the exploration. So, recommendations become more randomised.

It can be concluded that the results of the algorithm's performance depend heavily on the choice of ϵ value. The probability of obtaining only a sub-optimal solution both during either exploration or exploitation is still quite high. Such an unguided nature of bandit's behaviour is considered to be its heaviest drawback.

After the Algorithm (2) was applied to both realisations of LinUCB bandit. The results of the simulations were visualised and are depicted on the Figure (3.2). From the plots a conclusion can be drawn that when the value of α is assigned to incorporate more uncertainty in the expected reward the CTR is lower in both cases.

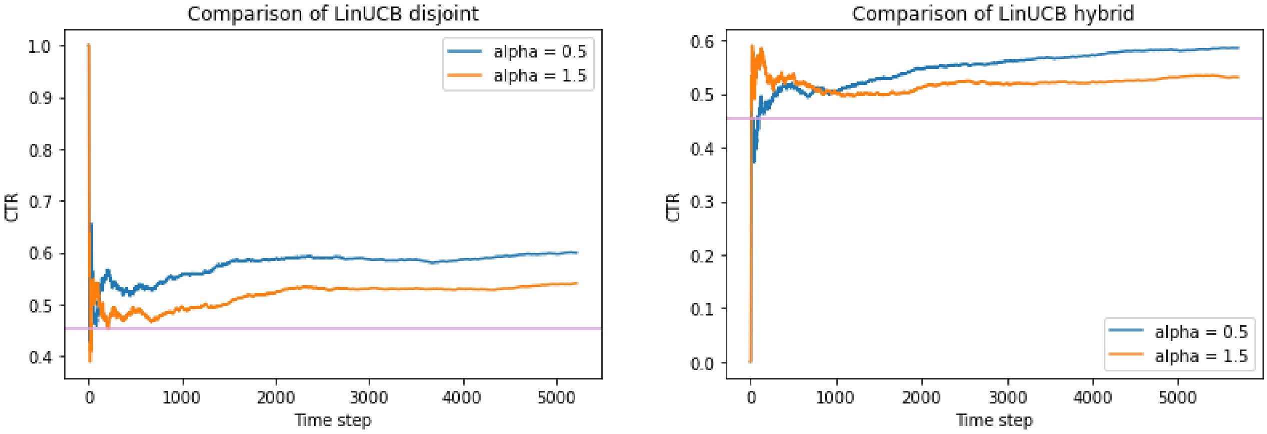

Figure 3.2: LinUCB disjoint and hybrid CTR simulations

When the emphasis is given on exploitation the results show that disjoint and hybrid version reach around 60% and 57% respectively. Meanwhile, with the value of $\alpha = 1.5$ the CTR value is approximately 53% for the first realisation and 51% for the second one.

After separate observation of algorithms performance their results were compared on one plot simultaneously:

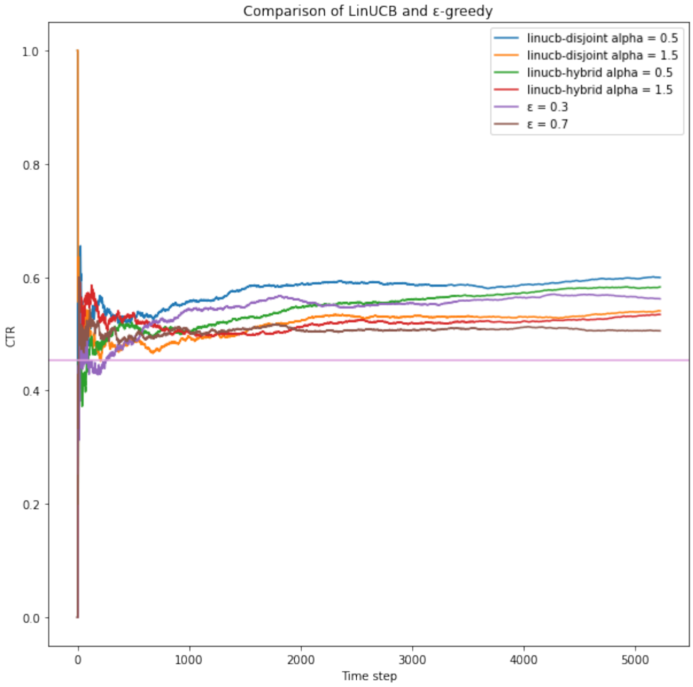

Figure 3.3: Comparison of $\epsilon - greedy$ and LinUCB algorithms

And the mean CTR values for each bandit were calculated:

$\epsilon - greedy$ CTR mean	
$\epsilon = 0.3$	$\epsilon = 0.7$
54.39%	50.57%

LinUCB disjoint CTR mean	
$\alpha = 0.5$	$\alpha = 1.5$
57.69%	51.79%

LinUCB hybrid CTR mean	
$\alpha = 0.5$	$\alpha = 1.5$
54.5%	51.78%

Table 3.2: Mean values of CTR for $\epsilon - greedy$ and LinUCB

By observing the obtained statistics, a conclusion can be drawn that the contextual bandits outperform the unguided implementation of MAB both in exploitation and exploration cases. These evidences prove the effectiveness of the LinUCB approach in comparison to the $\epsilon - greedy$ implementation.

4 Further Explorations

4.1 Thompson Sampling for Contextual Bandits

The idea of Thompson Sampling is one of the most basic proposed solutions alongside $\epsilon - greedy$ for the balance between exploitation and exploration. The trade-off is realised through the introduction of some prior distribution for each arm. In this case the mean serves as the expected reward and the standard deviation incorporates the uncertainty about a specific recommendation. The posterior is then calculated with the usage of prior and feedback obtained from the user. However, the disadvantage of such approach leads to a situation when after some time the algorithm always chooses the arm which has the winning distribution (i.e. brings the highest approximated reward)

The research in [1] proposed to add the effect of context in the distribution. This idea helps to overcome potential drawback of the initial approach. The incorporation is done with the calculation of the posterior which is described by:

$$P(\tilde{\mu}|r_{t,a_t}) \propto P(r_{t,a_t}|\tilde{\mu})P(\tilde{\mu}), \tag{4}$$

where $P(r_{t,a_t}|\tilde{\mu})$ is the likelihood function and $P(\tilde{\mu})$ is the prior.

The reward is approximated from the distribution $\mathcal{N}(\hat{\mu}, \nu^2 B^{-1})$. Looking at the mean, $\hat{\mu}$ is calculated with the multiplication of the covariance matrix of contexts B and vector emphasising the needed context f. In the standard deviation the variable $\nu = R\sqrt{\frac{24}{\epsilon} d \ln \frac{1}{\delta}}$ is a parameter which reflects what level of exploration is incorporated in the approximation.

On each step of the algorithm the $\tilde{\mu}$ is drawn from the assigned distribution. The choice of arm recommendation is based on the maximisation of $\hat{\mu} x_{t,a}$. If the drawn variant is approved by the Algorithm (2) the context updates the information stored in B and f. This way the problem of extensive exploitation of the original approach is fixed.

The described approach was tested with four combinations of different variables R, ϵ and δ in order to see their influence on the trade-off. The visualised results are presented at the Figure (4.1) with the corresponding mean CTR values in the Table (4.1).

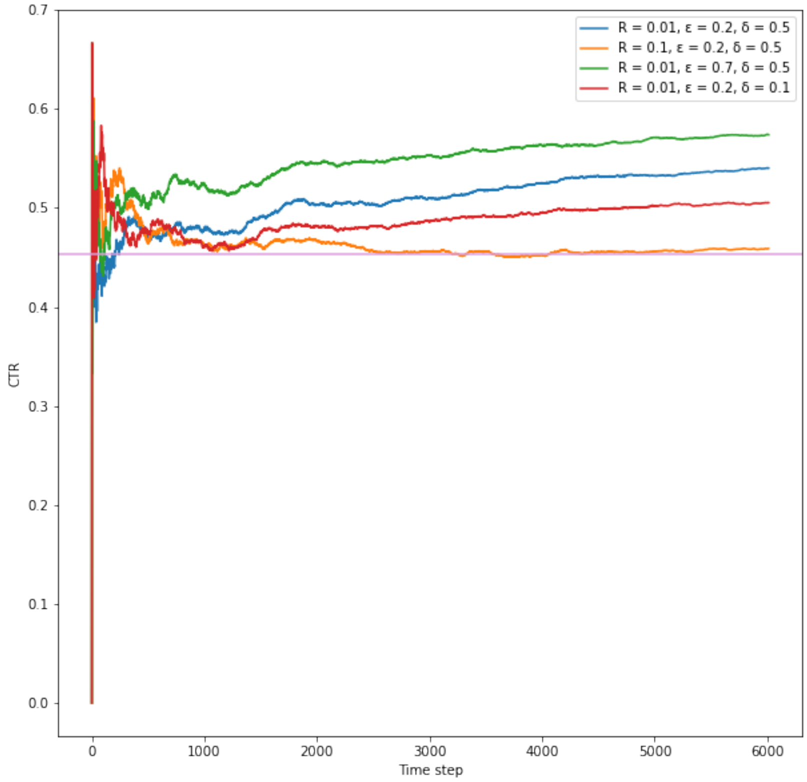

Figure 4.1: Comparison of $\epsilon - greedy$ algorithms

Blue line	Orange line	Green line	Red line
50.83%	46.27%	54.67%	48.82%

Table 4.1: Mean values of CTR for Thompson Sampling for Contextual Bandits

For the current section analysis the results of the "blue" simulation is assumed to be the base. The following conclusions can be made from the obtained results. The increase in the value of ϵ leads to a decrease in the standard deviation (uncertainty incorporated). Therefore, it can also be interpreted as more emphasis is given to exploitation. Consequently, the increase in R and δ parameters lead to exactly the opposite conclusion. Moreover, the change in the first parameter lead to much more randomised recommendations from the bandit.

After the internal comparison comes the analysis of performance among already realised algorithms:

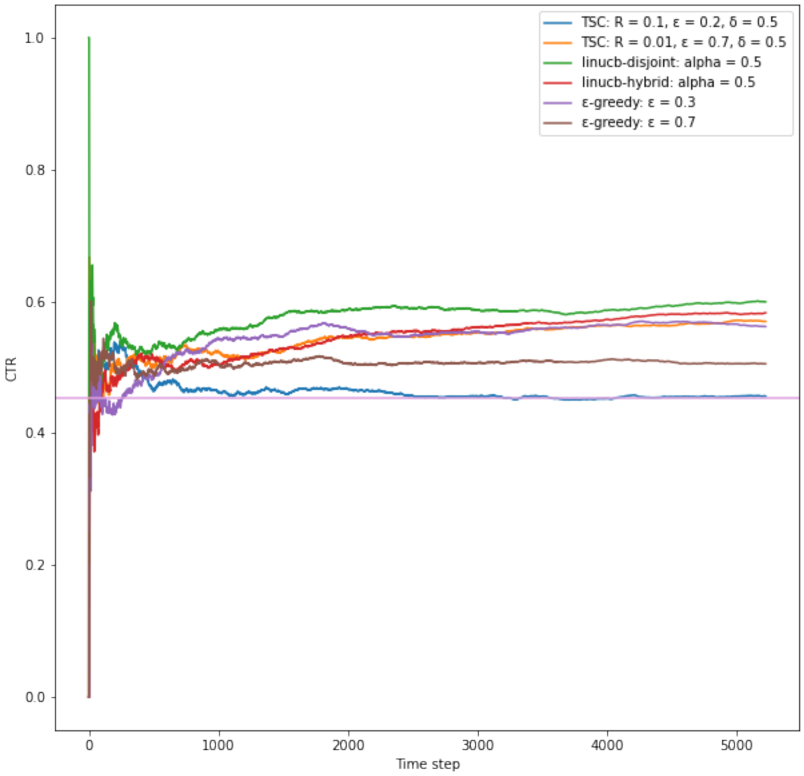

Figure 4.2: Comparison of $\epsilon - greedy$, LinUCB and "TSC" bandits

For simplification of visualisation and comprehension only a set of models were depicted. The obvious result on Figure (4.2) shows that the simulation of Contextual Thompson Sampling with $R = 0.1$ results in the worst performance among all. Interesting observation of the variant depicted as an orange curve. Firstly, its comparison with the $\epsilon - greedy$ algorithm which has more emphasis on exploitation reveals that the performance is quite similar. Although, some tendency of Thompson Sampling winning can be observed closer to the end. Secondly, the approach described in this section performs worse than both LinUCB implementations. Taking into consideration the online nature of the training more simulations must be performed to derive final conclusions. However, such result encourages the consideration of more efficient solution.

4.2 Neural Thompson Sampling

The development in the sphere of neural networks lead to the discovery of new algorithms for solving the MAB problem. One of the latest modifications is called Neural Thompson Sampling. This complex algorithm deserves to be reviewed more thoroughly by another research. This paper focuses only on explaining its basic idea with comparison of CTR simulation results.

The most impressive feature of this algorithm is that it moves from the linear relation of payoffs to the non-linear case. This is allowed with the introduction of a sequential neural network (multi-layer perceptron consisting of linear type and ReLU activation function) for every arm.

Let $f(x_{t,a}, \theta)$ define the process of applying each layer on some context $x_{t,a}$ with some weights θ. Also, let $g(x_{t,a}, \theta) = \nabla f(x_{t,a}, \theta)$, that is the gradient of the applied function. Finally, the matrix $U_0 = \lambda I_d$ denotes the covariance matrix with some regularisation parameter λ.

The idea of the authors basically comes down to the fact of approximation of the expected reward with the help of Taylor decomposition. It can be described as following:

$$r_{t+1,a} = f(x_{t,a}, \theta_t) + g^T(x_{t,a}, \theta_t)(\theta_t - \theta_0), \tag{5}$$

where θ_0 is the initially defined set of weights.

So, the approximation of reward at time step t for each arm is drawn from the distribution described as $\mathcal{N}(f(x_{t,a}, \theta_t), \nu^2 \sigma_t^2)$. Here the term ν is the level of uncertainty incorporated in the distribution which balances the trade-off. The parameter $\sigma^2 = \lambda g^T(x_{t,a}, \theta_{t-1}) U_{t-1}^{-1} g(x_{t,a}, \theta_{t-1})/m$ where m is the width of the network.

The arm is chosen if its approximated reward is maximised among all. The update $U_t = U_{t-1} + g(x_{t,a}, \theta_t) g^T(x_{t,a}, \theta_t)/m$ is done according to Algorithm (2). The used weights θ_t are assigned so that they minimize the following loss function:

$$L(\theta) = \sum_{i=1}^{t} (f(x_{i,a}, \theta) - r_{i,a})^2/2 + m\lambda ||\theta - \theta_0||_2^2/2. \tag{6}$$

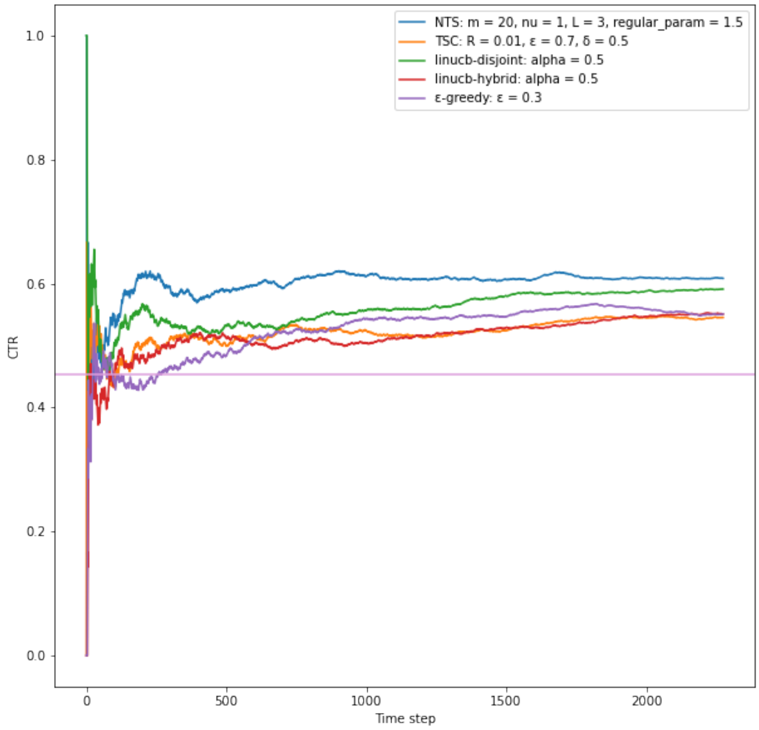

Figure 4.3: Comparison of "NTS" with other algorithms

The described algorithm was realised and compared only with the best results among all models. The results of the simulations are presented on Figure (4.3).

By observing the obtained values, it can be seen that Neural Thompson Sampling outperforms all of the models. It attains the result of 60% CTR. Nevertheless, some important aspects should be mentioned. To begin with, the simulation for this algorithm was only performed for one epoch. This is due to the fact that its training time takes twice as much as the training for 3 epochs of other algorithms. Even though $T \approx 2250$ is sufficient more simulations could be performed. Another thing to mention is that the neural network has a number of hyperparameters. Their influence on the results are to be explored. Finally, the experiment was seeded for the reproducibility of the results. However, the stability of the results needs to be studied thoroughly. Nonetheless, Neural Thompson Sampling has the potential to achieve greater results among both contextual and non-contextual type of multi-armed bandits.

5 Conclusion

Deriving the conclusion, this paper reached its goal in studying and comparing the efficiency of different type of contextual bandits. The selection of models offers a wide possibility of implementation. Also, the application of the CTR metric allows to obtain more empirical along with theoretical results. Taking into consideration the shortcomings, this paper could have regard more topics for study like the case with non-fixed number of arms or the problem of cold-start. Nonetheless, current study offers a solid foundation for the exploration in the field. In addition to the listed topics of interest, further studies could be concentrated around the idea proposed in Section (4.2).

References

[1] Shipra Agrawal and Navin Goyal. “Thompson sampling for contextual bandits with linear payoffs”. In: *International conference on machine learning*. PMLR. 2013, pp. 127–135.

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

[3] Donald A Berry and Bert Fristedt. “Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)”. In: *London: Chapman and Hall* 5.71-87 (1985), pp. 7–7.

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

[5] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. “Neural thompson sampling”. In: *arXiv preprint arXiv:2010.00827* (2020).