Faculty of Computer Science

Data Science & Business Analytics

Moscow 2024

Research project:
“Contextual Bandits for Recommendation Systems“
“Контекстуальные бандиты для рекомендательных систем“

Author: Gorelskii Roman, group #БПАД212

Project Supervisor: Samsonov Sergey, Research Fellow, Faculty of Computer Science, HSE University

Table of contents

01

Introduction

02

Problem statement

03

MovieLens data

04

Methods of realisation

05

Simulations and results

06

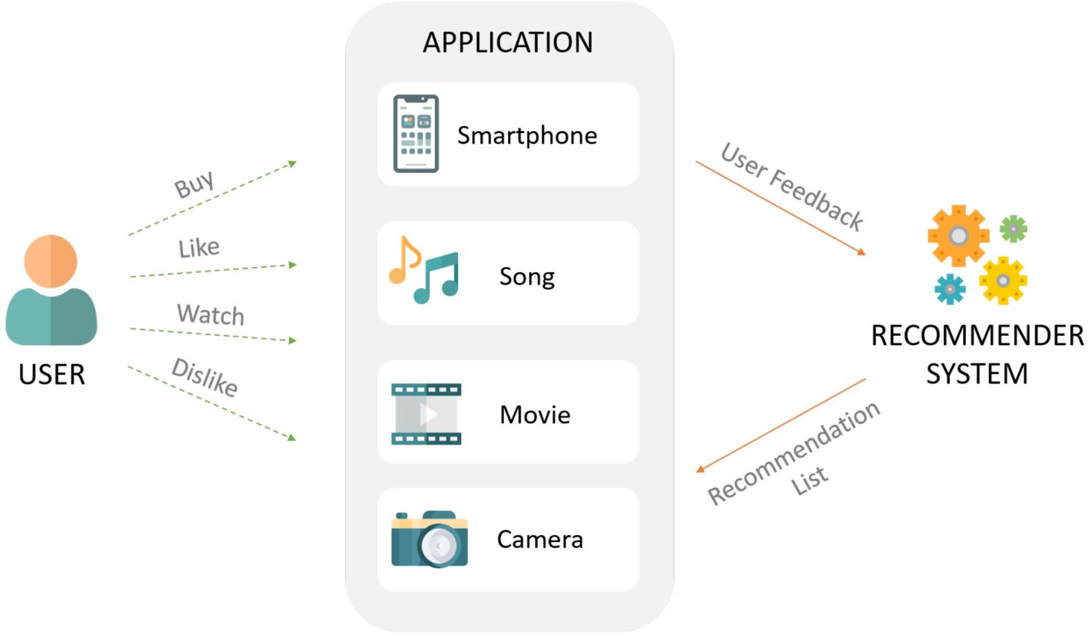
Conclusions and prospects

01

Introduction

Recommender system is a reinforcement learning based algorithm which is aimed at the analysis of big data providing users with more relevant products for them.

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 solution is defined by the multi-armed bandit (MAB) problem. The algorithm’s strategy is to select the most optimal variant (arm) for recommendation. It allows to reduce the computational complexity with the possibility to test the performance on the traffic recorded previously.



Recommender system performance visualised
Source: https://www.mdpi.com/2076-3417/10/16/5510?h=1

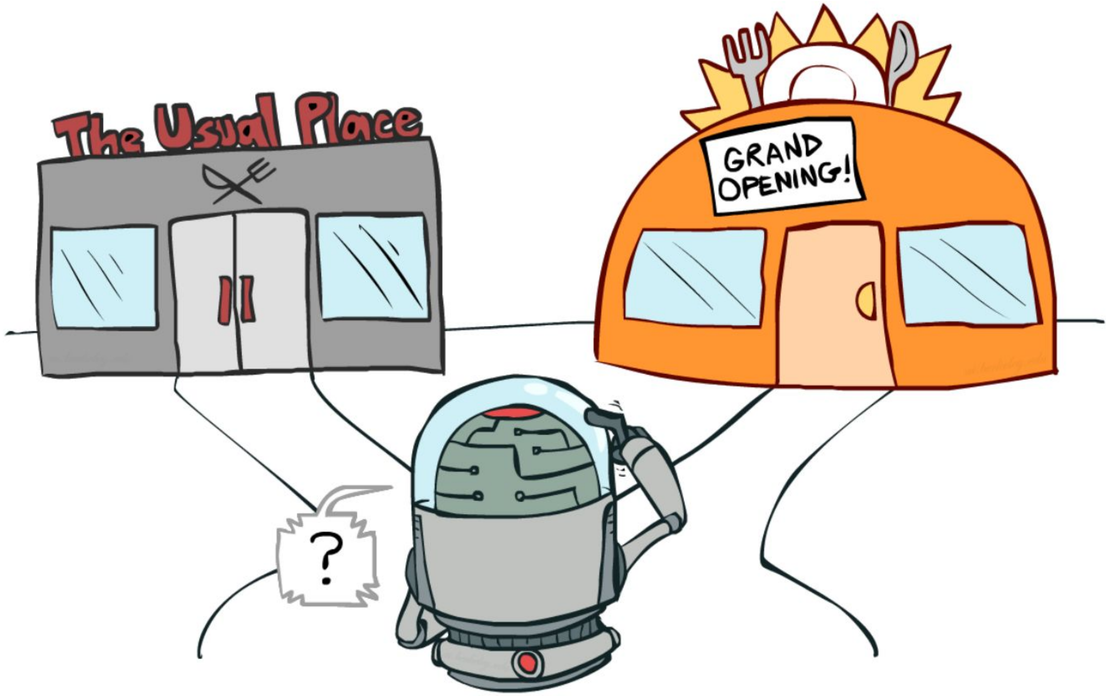
02

Problem statement

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.



Intuitive visualisation of dilemma concern
Source: https://lilianweng.github.io/posts/2018-01-23-multi-armed-bandit/

02

Problem statement

Mathematical formulation of the problem

The goal of developing a recommender system is to design the algorithm in such a way so that the regret is minimised with respect to the optimal arm-selection. The regret is defined formally from the eq. (1), where rt,at∗ is defined as the maximum received payoff from choosing an optimal arm, rt,at is the total T-trial payoff.

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

Formula for the regret calculation

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) which is formally defined by the eq. (2).

$$CTR_t = \frac{CumRew_t}{t}. \quad (2)$$

Formula for the click-through rate calculation at time step t

02

Problem statement

Objectives to accomplish

This research focuses on the objectives to explore three different approaches of MAB implementations:

- **ε-greedy** - one of the basic simple algorithms selected as a benchmark. Its realisation is a good starting point for better understanding of the internal structure of the bandit.
- **LinUCB** - the main algorithm on which the current research is focused on. It helps to explore how the incorporation of the context information helps to improve the performance.
- **Contextual Thompson sampling** - modification of one of the basic MAB algorithms which pushes the research further with exploration of alternative considerations and introduction of non-linearity for the payoffs.

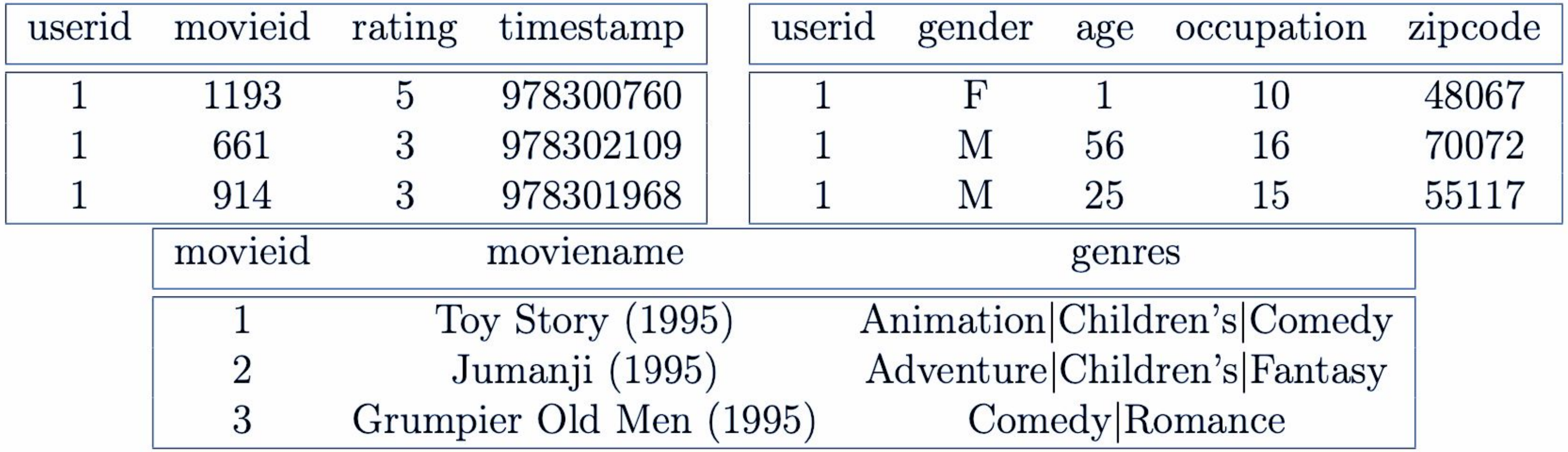
Each of the algorithm is studied in terms of discussing the underlying ideas behind the implementation as well as performing numerical simulations with results comparison.

03

MovieLens data

Data description

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

First three rows of “ratings” (top left dataset), “users” (top right dataset) and “movies” (bottom dataset)

03

MovieLens data

Data preprocessing

For the purpose of correct CTR simulations the following preprocessing steps were performed with the data:

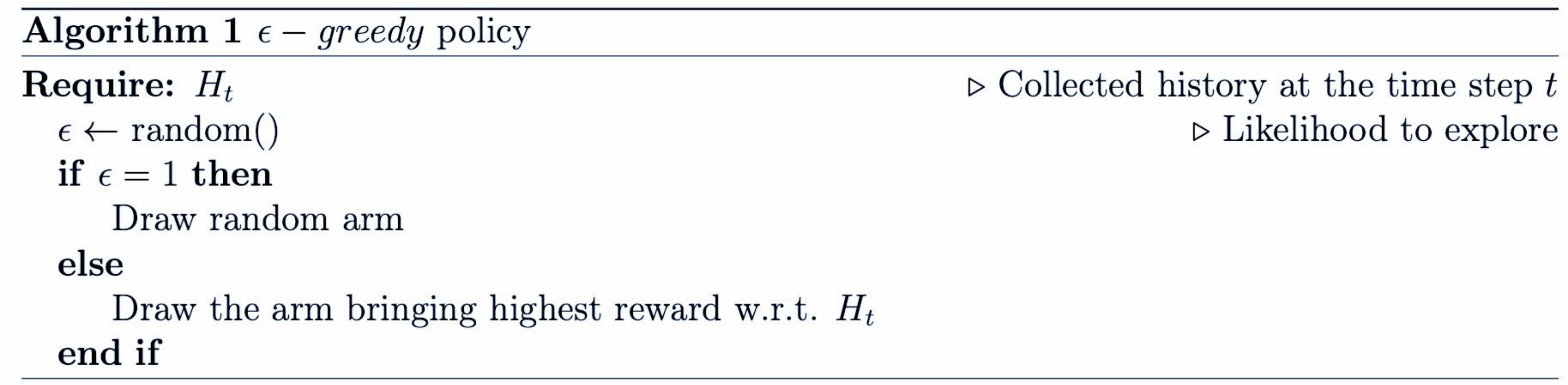
- “users”
  - one-hot encoding for “gender” category
  - encoding of “occupation” category
- “movies”
  - encoding of “genres” category
- “ratings”
  - leaving only 30 most often rated movies
  - conversion of ratings with “5” to the equivalent of a click (“1”); others are not considered as click (“0”)
  - shuffling the rows for the purpose of online learning environment imitation

04

Methods of realisation

ε-greedy algorithm

One of the most basic algorithms used to solve the multi-armed bandit problem is called ε-greedy. This implementation aims at providing the policy which allows to balance between the exploration and exploitation through the introduction of ε likelihood variable. It 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.



```
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
```

The algorithmic description of ε-greedy algorithm

Such an unguided implementation emphasizes the improvements that are brought by other algorithms regarded in this paper.

04

Methods of realisation

LinUCB algorithm

To provide the intuitive explanation of LinUCB algorithm the following notations must be introduced:

$x_{t,a}$ - context for the user

$\mathbf{z}_{t,a}$ - context for the recommended object

D_a - design matrix with m rows of contexts that are observed previously

$\mathbf{c}_a$ - response vector

I_d - identity matrix

$\hat{\theta}_a$ - estimated coefficient

04

Methods of realisation

LinUCB disjoint algorithm

The intuitive interpretation of the upper confidence bound is that it serves as a form of a confidence interval for the expected payoff which allows to incorporate the level of uncertainty about the obtained reward.

The arm selection policy is defined by the eq. (3).

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

$\hat{\boldsymbol{\theta}}_a = (\mathbf{D}_a^\top \mathbf{D}_a + \mathbf{I}_d)^{-1} \mathbf{D}_a^\top \mathbf{c}_a$ - estimation by ridge regression

$x_{t,a}^T \hat{\theta}_a$ - expected payoff of an arm a

$\mathbf{A}_a = \mathbf{D}_a^\top \mathbf{D}_a + \mathbf{I}_d \longrightarrow \sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}$ - estimated standard deviation

α - coefficient regulating level of uncertainty incorporated

04

Methods of realisation

LinUCB hybrid algorithm

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. For example, recommending movies of a particular genre to similar users might bring higher reward. This assumption can be reflected mathematically through the introduction of features in the expected payoff.

$$E[r_{t,a}|x_{t,a}] = z_{t,a}^T \beta^* + x_{t,a}^T \theta_a^*$$

Hybridity of the implementation consists in the fact that the coefficient stored at β* are shared among all arms.

The introduction of arms’ similarities also influences the estimated standard deviation.

04

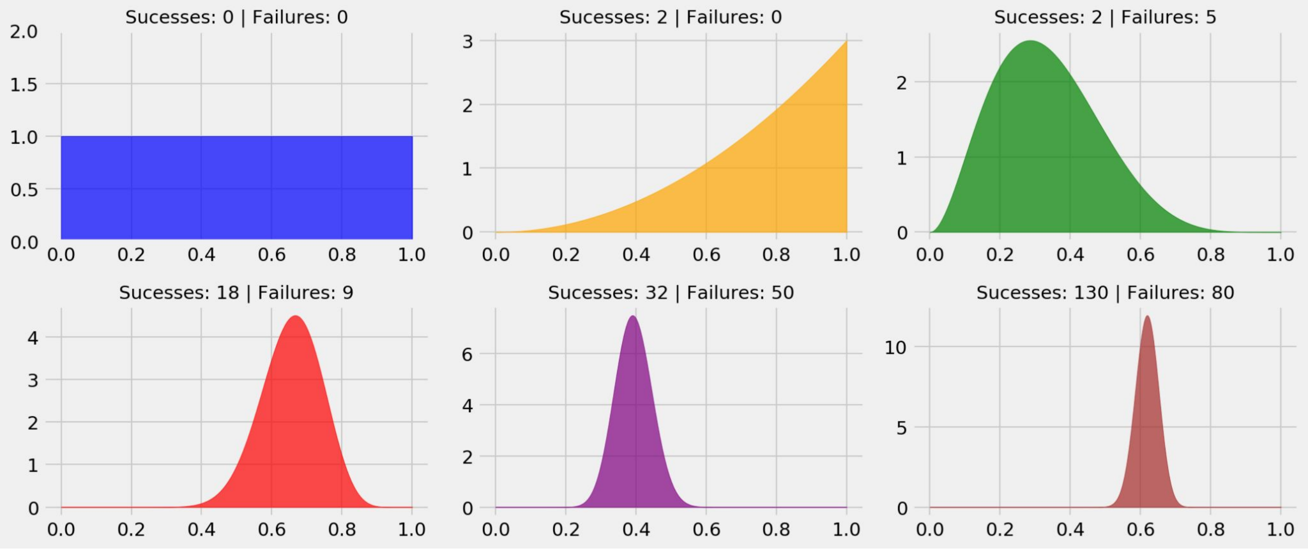
Methods of realisation

Thompson sampling

The idea of Thompson Sampling is one of the most basic proposed solutions alongside ε-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.

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)



Visualisation of Thompson sampling results through time
Source: https://gdmarmerola.github.io/ts-for-bernoulli-bandit/

04

Methods of realisation

Contextual Thompson sampling

The proposition 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 the eq. (4).

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

$P(r_{t,a_t}|\tilde{\mu})$ - likelihood function

$P(\tilde{\mu})$ - prior defined for an arm

04

Methods of realisation

Contextual Thompson sampling

Further approximation of the reward is done by drawing it from the following assigned distribution: $\mathcal{N}(\hat{\mu}, \nu^2 B^{-1})$

B - covariance matrix of contexts

f - vector which emphasizes the needed context

$\hat{\mu}$ - expected reward from the multiplication of B and f

$\nu = R\sqrt{\frac{24}{\epsilon} d \ln \frac{1}{\delta}}$ - parameter reflecting the level of exploration incorporated in the approximation

The arm selection policy is concluded in the maximisation of the product between the approximated reward and context.

04

Methods of realisation

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.

Here are the notations that will be used in the explanation of the idea that stands behind the algorithm:

$f(x_{t,a}, \theta)$ - process of applying layers on the context

θ - set of weights

$g(x_{t,a}, \theta) = \nabla f(x_{t,a}, \theta)$ - gradient of the applied function

$U_0 = \lambda I_d$ - covariance matrix of contexts with regularization parameter λ

04

Methods of realisation

Neural Thompson sampling

The idea of the approximation of the expected reward consists in the usage of Taylor decomposition. It is described by the eq. (5).

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

At some time step the approximated reward for each arm is drawn from the following distribution: $\mathcal{N}(f(x_{t,a}, \theta_t), \nu^2\sigma_t^2)$

ν - parameter incorporating the level of uncertainty

$\sigma^2 = \lambda g^T(x_{t,a}, \theta_{t-1})U_{t-1}^{-1}g(x_{t,a}, \theta_{t-1})/m$ - approximated variance of the reward

m - width of the network

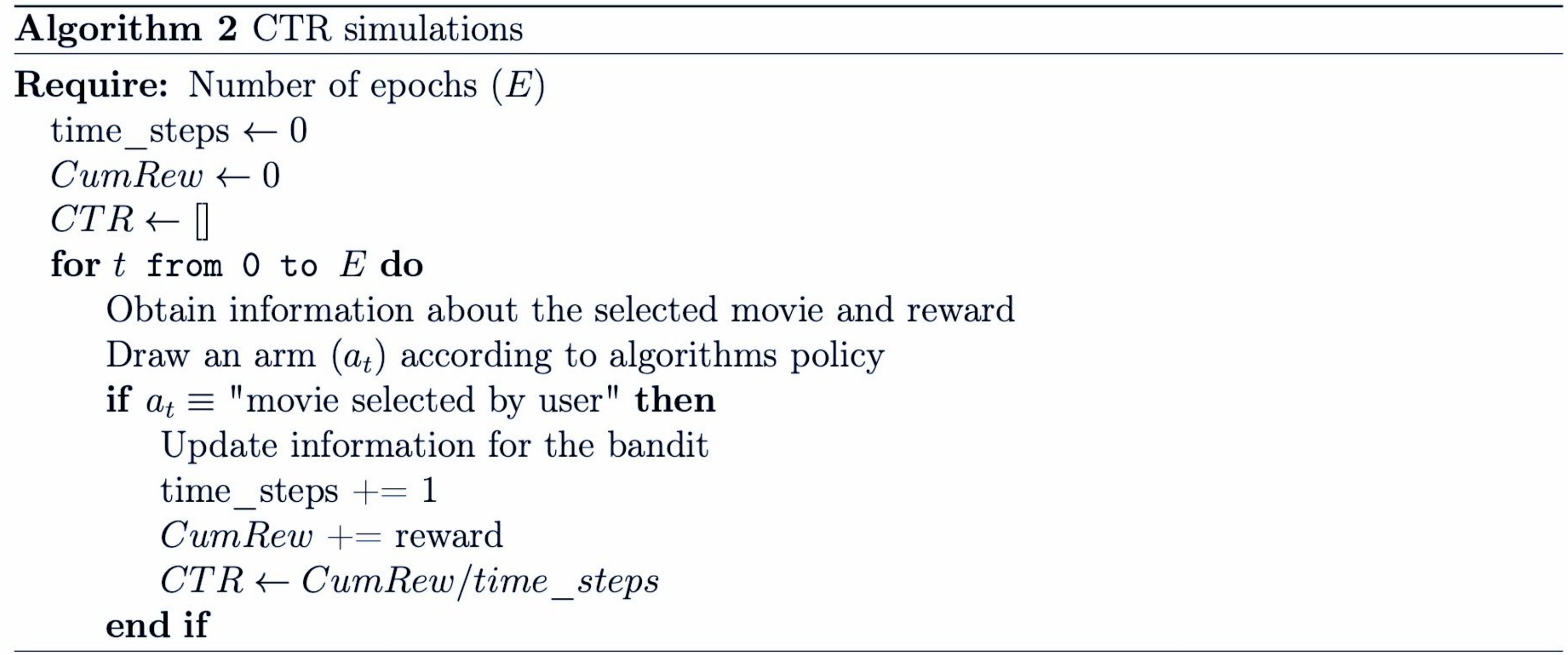
The updated weights are chosen to minimise the loss function: $L(\theta) = \sum_{i=1}^{t}(f(x_{i,a}, \theta) - r_{i,a})^2/2 + m\lambda||\theta - \theta_0||_2^2/2. \ (6)$

The arm selection policy is the one that maximizes the approximated reward.

05

Simulation and results

Click-through rate simulations



Algorithm 2 CTR simulations

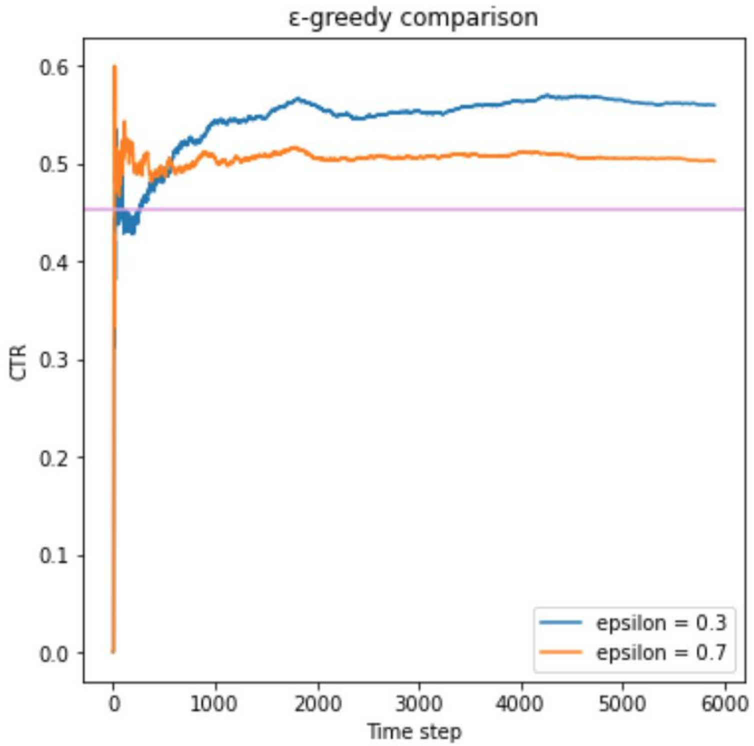
```
Require: Number of epochs ($E$)
  time_steps $\leftarrow 0$
  $CumRew \leftarrow 0$
  $CTR \leftarrow []$
  for $t$ from 0 to $E$ do
      Obtain information about the selected movie and reward
      Draw an arm ($a_t$) according to algorithms policy
      if $a_t \equiv$ "movie selected by user"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

The algorithmic description of click-through rate simulations applied to the data

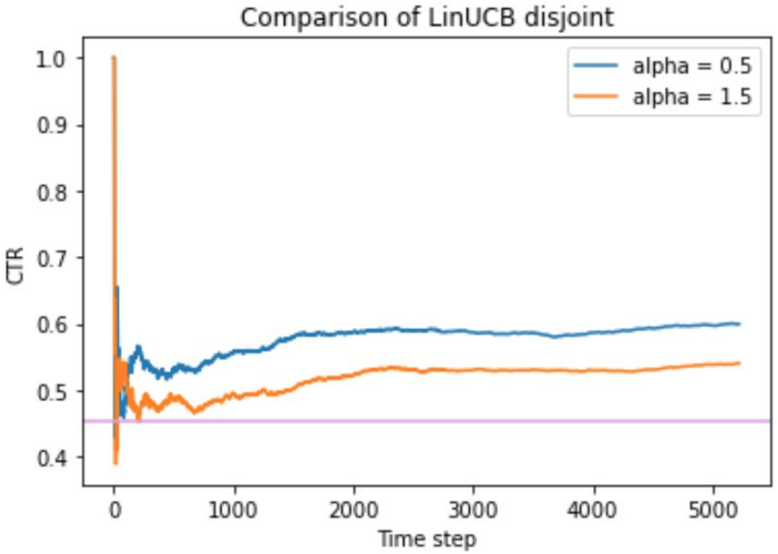
05

Simulation and results

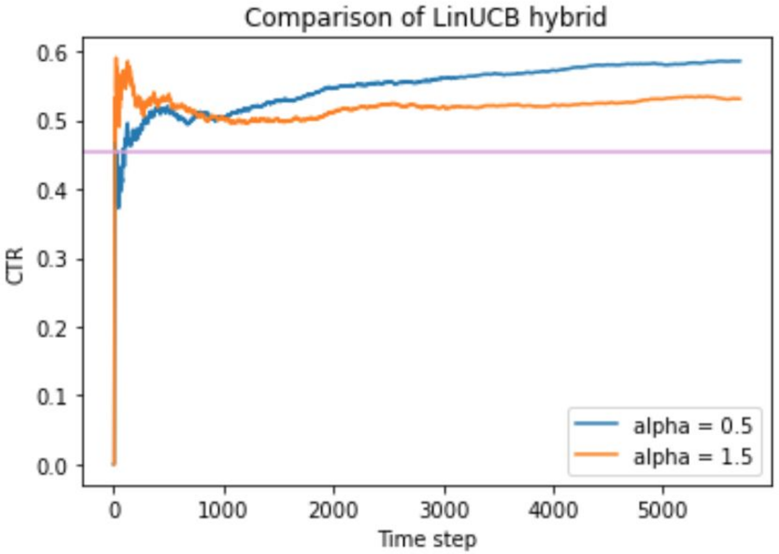
ε-greedy simulations



LinUCB disjoint simulations

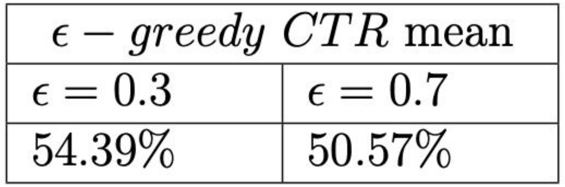


LinUCB hybrid simulations



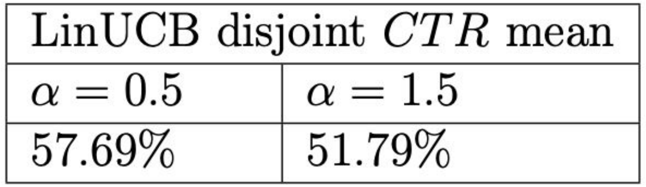
05

Simulation and results



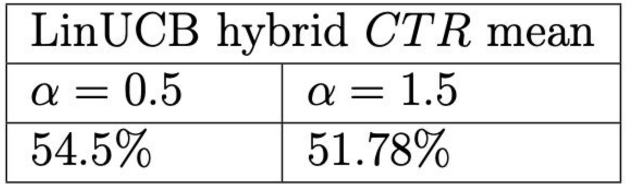
$\epsilon - greedy\ CTR$ mean	
$\epsilon = 0.3$	$\epsilon = 0.7$
54.39%	50.57%

Mean values of CTR for ε-greedy



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

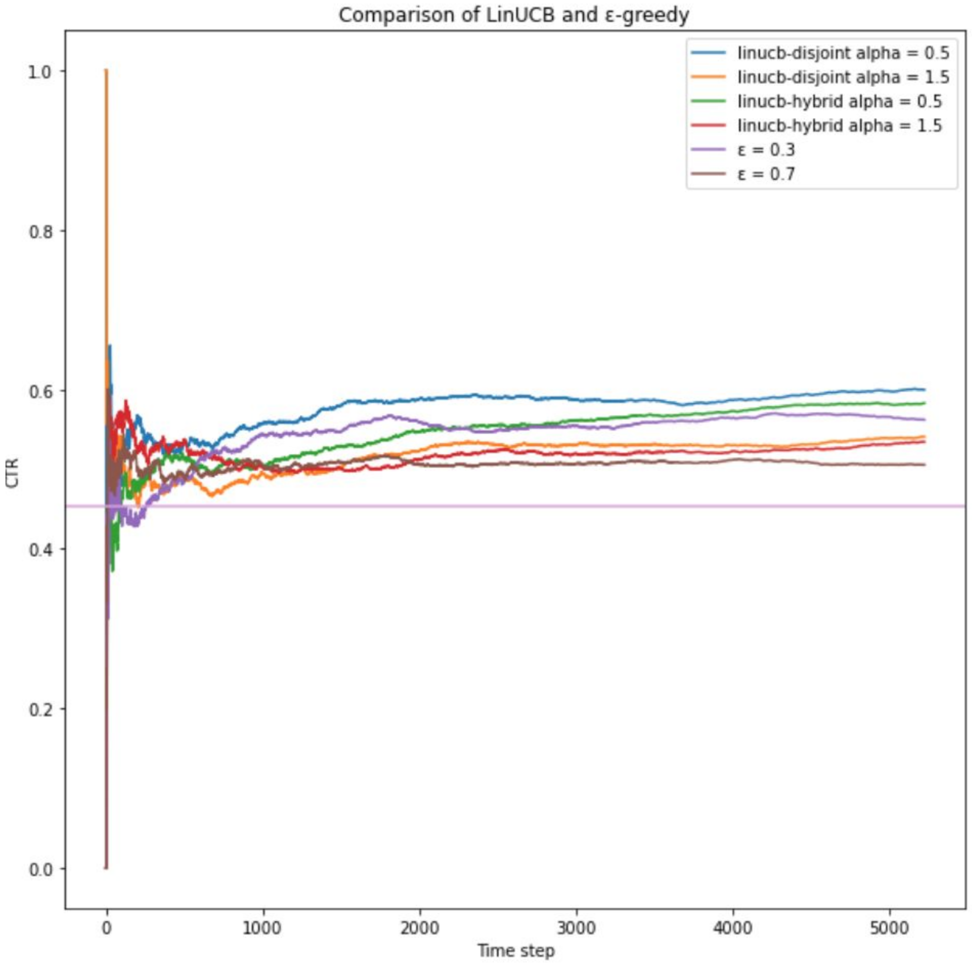
Mean values of CTR for LinUCB disjoint



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

Mean values of CTR for LinUCB hybrid

Comparison of ε-greedy and LinUCB simulations



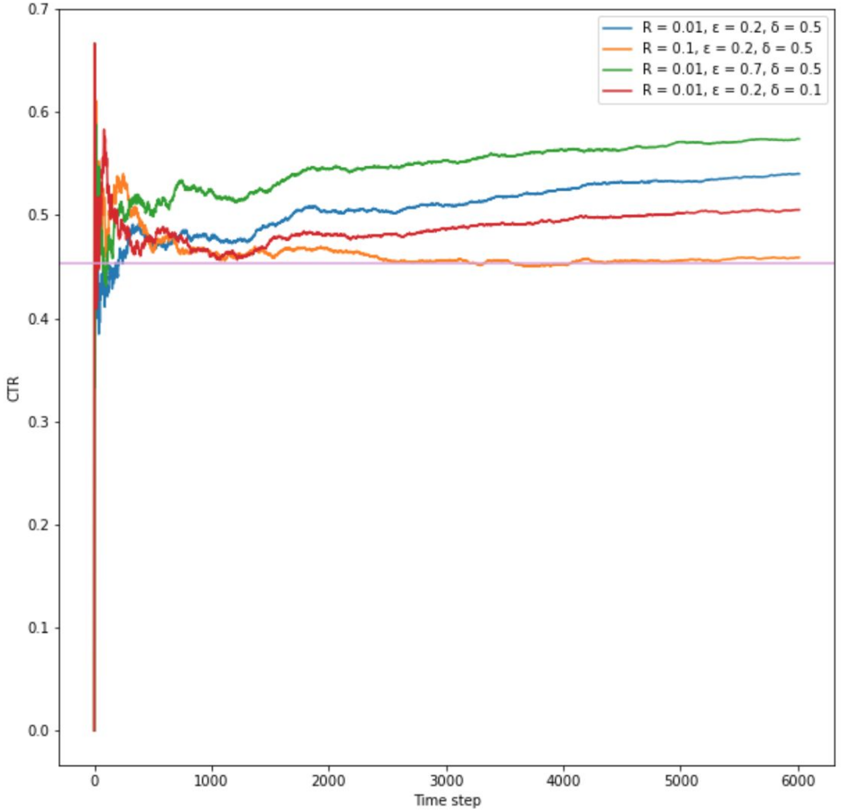
05

Simulation and results

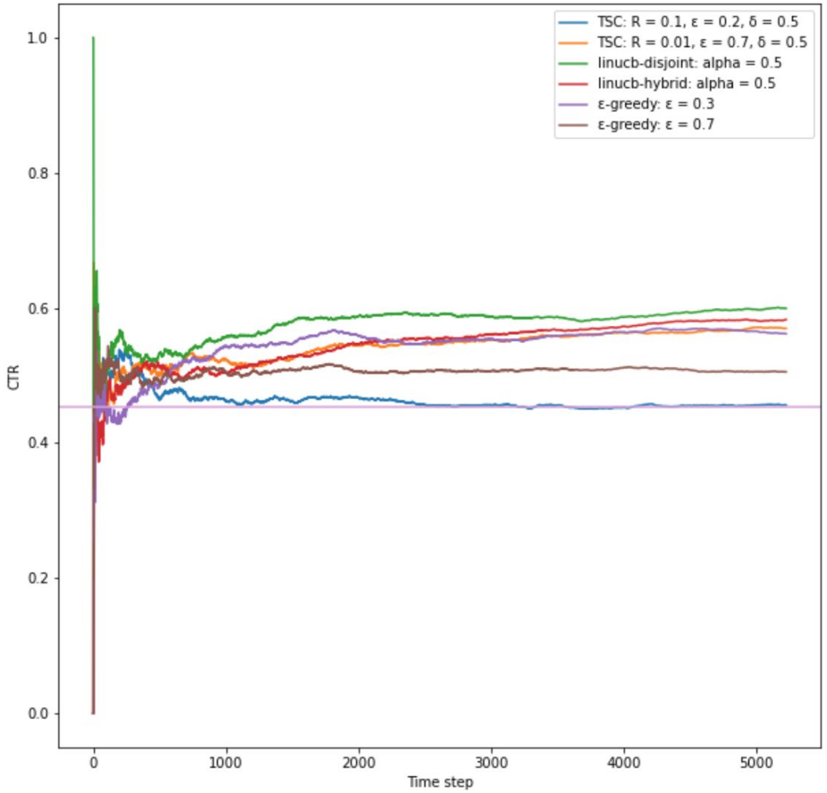
Blue line	Orange line	Green line	Red line
50.83%	46.27%	54.67%	48.82%

Mean values of CTR for contextual Thompson sampling

Contextual Thompson sampling simulations



Its comparison with ε-greedy and LinUCB



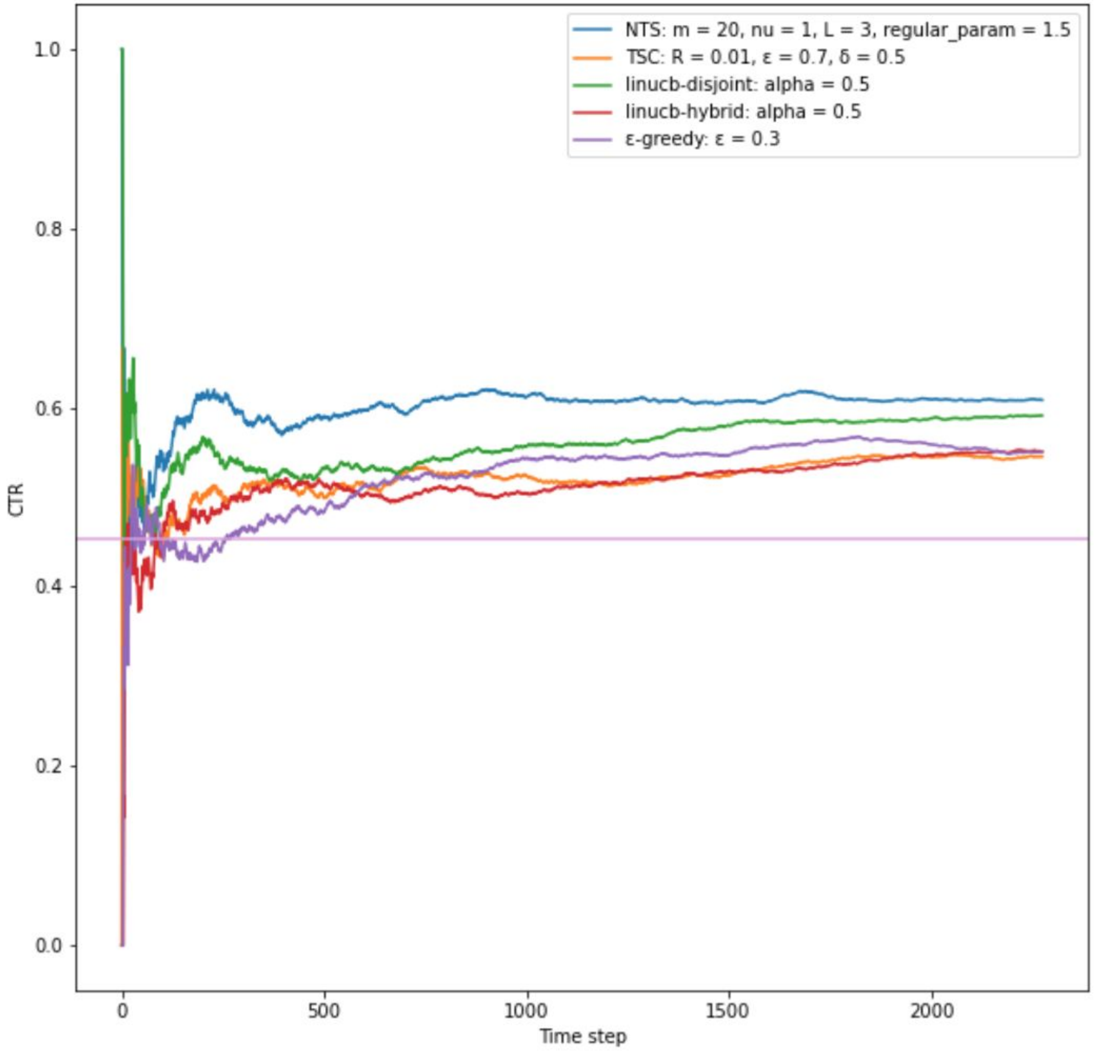
05

Simulation and results

Neural Thompson sampling comparison with other algorithms

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. 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 stability of the results needs to be studied thoroughly even though the experiments were seeded. Nonetheless, Neural Thompson Sampling has the potential to achieve greater results among other realised algorithms.



06

Conclusions and prospects

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 the empirical comparison of algorithms alongside the theoretical intuition standing behind them.

The future research can be aimed at several further explorations. Firstly, even though the current study offers a solid foundation for introduction and exploration to the field of multi-armed bandits many of the topics disregarded. Possible innovations could be discovered by studying the case with non-fixed number of arms or the problem of cold-start. Secondly, as it was mentioned the exploration of stability and performance under various hyperparameters of Neural Thompson sampling is to be studied. So, further studies could be concentrated around those ideas.

Source:
https://www.pinterest.com/pin/brc-global-standard-for-food-safety-issue-7-audit-checklist--412009065903328623/

Reference list

[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).

Thank you for your attention!
We can start the discussion now