DRL for recommendation

Yuefan Wu
PB18000149

April 24, 2021

1 Background

The majority of online bidding are based on offline optimization algorithms, which is challenging when the environment is unstable.
RL based on online ad techinques can update strategy continuously.
However, most works focus only on the revenue of ads. The negative influence of ads is ignored.

2 Introduction

Designing an appropriate advertising strategy is a challenging problem,since (i) displaying too many ads or improper ads will degrade user experience and engagement; and (ii) displaying insufficient ads will reduce the advertising revenue of the platforms.
Three parts are considered:
1. whether to interpolate an ads
2. which ads to be shown
3. where the ads to be interpolated

And these three parts are internally related.

3 Compare

Traditional is shown as in Figure 1:
First one takes the state space and outputs Q-values of all locations.(Can determine the optimal location)
Second one inputs a state-action pair and outputs the Q-value corresponding to a specific action.(Can select a specific ad)
The new one is shown as Figure 2:
The first is the novel Q-network architecture and the second one is the detailed one.
As shown in Figure 2, the novel network get state-action pair as inputs and outputs the Q values. The $V(S_t)$ on the left of the detailed network is the result from state, cause the Q values should be tightly connected to the state.
While the advantage function $A(s_t, a_t)$ is due to there must be influence both from the state and

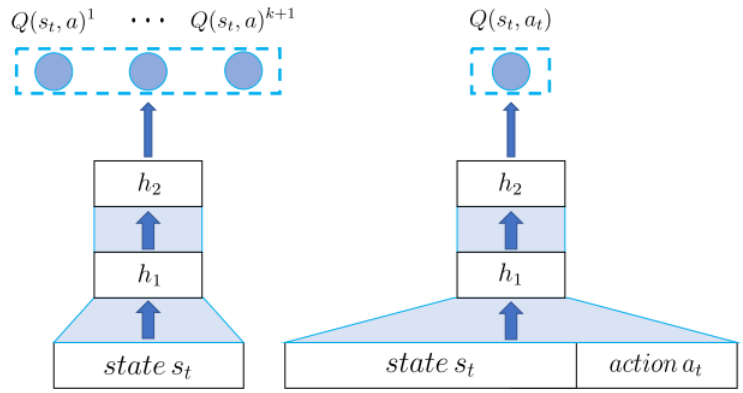


Figure 1: Traditional

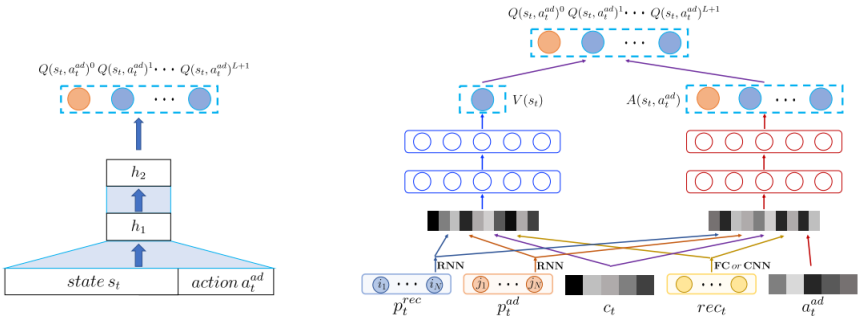


Figure 2: New

from the action taken.

4 Proposed Framework

The proposed framework is the new one shown in Figure 2.
Challenges faced:

- Three sub-actions shown above
- The candidate ads and locations are interactive to maximize the reward.
- The advertise agent should simultaneously maximize the income and minimize the negative influence

4.1 Process of State and Action

s_t consists of a user's rec/ads browsing history, the contextual information and rec-list of current request.
Leverage two GRUs, one to capture users' sequential preference of recommendation(comments); the other to capture the ads
The outputs of GRUs will be p_t^{rec} and p_t^{ad}
The contextual information c_t including app version and feed type(swiping up/down)
The rec-list $rec_t = \tanh(W_{rec}\text{concat}(rec_1, rec_2, ..., rec_L)+b_{rec})$
Representation of state s_t:

$$s_t = concat(p_t^{rec}, p_t^{ad}, c_t, rec_t) \tag{1}$$

Reward function:
income of ad r_t^{ad}
User experience r_t^{ex}:

$$r_t^{ex} = \begin{cases} 1 & continue \\ -1 & leave \end{cases}$$

$$r_t(s_t, a_t) = r_t^{ad} + \alpha r_t^{ex} \tag{2}$$

The network considers both the ads itself and the relevance between the ads and users, so the V corresponds to the ads itself while the A corresponds to the relevance.

4.2 Optimization

Optimize by minimizing a sequence of loss functions L(θ):

$$L(\theta) = E_{s_t, a_t, r_t, s_{t+1}}(y_t - Q(s_t, a_t; \theta))^2 \tag{3}$$

where $y_t = E_{s_t+1}[r_t + \gamma \max_{a_t+1} Q(s_{t+1}, a_{t+1}; \theta^T)|s_t, a_t]$
And the derivatives of loss funtion L(θ) is

$$\nabla_\theta L(\theta) = E_{s_t, a_t, r_t, st+1}(y_t - Q(s_t, a_t; \theta))\nabla_\theta Q(s_t, a_t; \theta) \tag{4}$$

And the meaning of y_t is the same as mentioned above.

4.3 The DQN architecture

Handle the second and third problems simultaneously.
It will get both s_t and a_t as input, which means the output will contain both the optimal location and the optimal ads.
And for the first problem, whether to interpolate an ads, there creates a new Q value called $Q(s_t, a_t)^0$ indicating that no ads should be interpolated in the current rec-list.
Besides, the Q value is divided into $V(s_t)$ which is determined by the state features, and the $A(s_t, a_t)$ which is determined by both state and action features.
The algorithm is shown in algorithm 1:

Algorithm 1: Off-policy Training of DEAR Framework

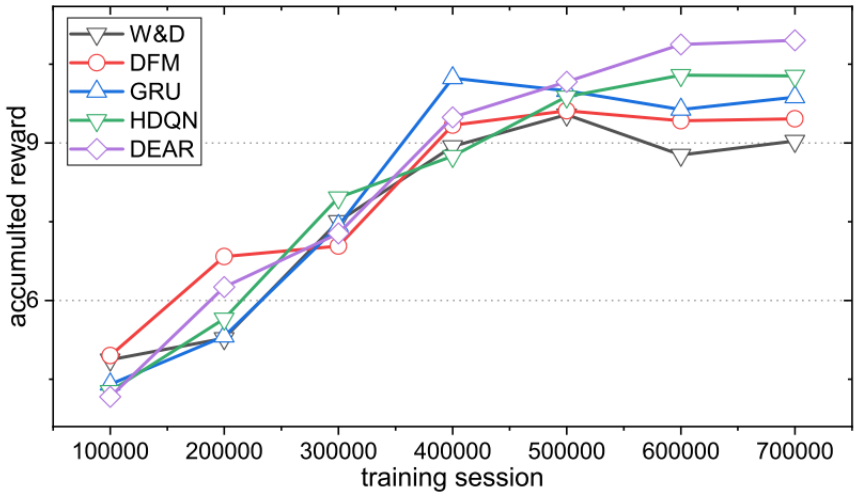
Initialize the capacity of replay buffer D
Initialize action-value function Q with random weights
for *session = 1 to M* **do**
  Initialize state s_0 from previous sessions
  for *t = 1 to T* **do**
    Observe state $s_t = concat(p_t^{rec}, p_t^{ad}, c_t, rec_t)$
    Execute action a_t following off-policy $b(s_t)$
    Calculate reward $r_t = r_t^{ad} + \alpha r_t^{ex}$ from offline log
    Update state to s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) into the replay buffer D
    Sample mini-batch of transitions (s, a, r, s') from the replay buffer D
    Set y =

$$\begin{cases} r & terminals' \\ r + \gamma \max_{a'} Q(s', a'; \theta) & non-terminals' \end{cases}$$

    Minimize(y-Q(s,a;θ))$^2 accordingtooptimizationequation$
  end
end

5 Experiments

No public data set is available, so the authors used dataset of March,2019 collected in a short video site, in which there are two types of videos ,i.e. normal videos and ad videos. And there are features like: id, like score, finish score, comment score. Result from paper:



6 New Ideas

1. Change the reward function r_t^{ex} into continuous value
   Changing the reward function from a discrete value into a continuous value has several considerations.
   First, the action that user is browsing the ads is a time-length action so it should not be easily represented by a binary value.
   Second, the action leave and stay cannot easily represent the attitude of the user.They may leave for search and come back later.
   The idea is to mapping the ratio of ads video that the user watched to -1,1. And if the user stay for a long time, it should get a positive reward, cause the user seem like it. However, if the user slip away, that means it doesn't get the user's interest.

2. Use Neural Graph Collaborative Filter to improve the expressive power of embedding.
   Those features mentioned above may have potential correlations. And NGCF can lead to the expressive modeling of high-order connectivity in user-item graph, effectively injecting the collaborative signal into the embedding process in an explicit manner.

3. Change the model to a more advanced one, like Double DQN, or Dueling DQN.

7 Reference

1. Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiaobing Liu, Xiwang Yang, Jiliang Tang. 2019 Deep Reinforcement Learning for Online Advertising in Recommender Systems. In AAAI.

2. Xiang Wang,Xiangnan He, Meng Wang, Fuli Feng, Tat-Seng Chua. 2019 Neural Graph Collaborative Filter. In SIGIR