DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems

PaperID: 4386

Xiangyu Zhao¹, Changsheng Gu², Haoshenglun Zhang²
Xiwang Yang², Xiaobing Liu², Hui Liu¹, Jiliang Tang¹

¹: Data Science and Engineering Lab, Michigan State University
²: Bytedance
Recommender Systems

- Assisting users in their information-seeking tasks
  - Goal: suggesting items that best fit user’s preferences

Music
- Netflix
- TikTok
- Amazon
- JD.COM

Video
- YouTube
- Kwai
- Taobao
- JD.COM

Ecommerce
- Amazon
- Taobao
- JD.COM

News
- CNN
- FOX NEWS

Social Friends
- Facebook
- LinkedIn
- 美团
- Yelp

Location based

Online Ads
- Google
- Dianping

Online App
- Huawei

Content
Advertising in Recommender Systems

- **Goal:** maximizing the advertising revenue from advertisers
- Assigning the right ads at the right place to the right consumers
Online Advertising Challenges

- Offline and static optimization

Guaranteed delivery

Real-time bidding
Online Advertising Challenges

- Reinforcement learning based online advertising

- Challenges:
  - Advertising revenue
  - User experience

VS
An Example of Online Advertising Impression

- Three tasks
  - Interpolate an ad?
  - The optimal location?
  - The optimal ad?

- Goals of ad agent
  - Maximizing advertising revenue
  - Minimizing the negative influence of ads on user experience
Definition

- **Markov Decision Process (MDP)**
  - Advertising agent interacts with environment (users)

- **State space $S$:**
  - A state $s_t \in S$ is defined as a user’s browsing history before time $t$ and the information of current request at time $t$

$$s_t = \text{concat}(p_{t}^{rec}, p_{t}^{ad}, c_{t}, rect)$$

- **Action space $A$:**
  - The action $a_t \in A$ is to determine three internally related tasks: interpolate an ad? the optimal location? the optimal ad?
Definition

- **Reward R:**
  - Income of ad \( r_{t}^{ad} \)
  - Influence of an ad on the user experience \( r_{t}^{ex} \)

\[
 r_{t}(s_{t}, a_{t}) = r_{t}^{ad} + \alpha \cdot r_{t}^{ex}
\]

- **Transition probability \( P \):**
  The state transition from \( s_{t} \) to \( s_{t+1} \) after taking the action \( a_{t} \)

\[
p(s_{t+1}|s_{t}, a_{t}, ..., s_{1}, a_{1}) = p(s_{t+1}|s_{t}, a_{t})
\]

- **Discount factor \( \gamma \):**
  - Discount factor \( \gamma \in [0,1] \) is introduced to measure the present value of future reward

\[
r_{t}^{ex} = \begin{cases} 
1 & \text{continue} \\
-1 & \text{leave} 
\end{cases}
\]
Classic DQN Architectures

- **Assumptions**
  - There are $|A|$ candidate ads for each request
  - The length of the rec-list is $L$
Novel DQN Architecture

- Three tasks
  - Task 1: Interpolate an ad?
  - Task 2: The optimal location?
  - Task 3: The optimal ad?
The first individual DQN architecture that can simultaneously evaluate the Q-values of multiple levels’ related actions.
Experimental Settings

- **Dataset from the short video app Douyin**

  Table 1: Statistics of the Douyin video dataset.

<table>
<thead>
<tr>
<th>session time</th>
<th>user</th>
<th>normal video</th>
<th>ad video</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.980 min</td>
<td>188,409</td>
<td>17,820,066</td>
<td>10,806,778</td>
</tr>
</tbody>
</table>

- **Metric**: accumulated reward of a recommendation session
Overall Performance Comparison

- Baselines
  - Wide & Deep
  - DeepFM
  - GRU4REC
  - Hierarchical DQN

Table 2: Overall performance comparison.

<table>
<thead>
<tr>
<th>method</th>
<th>reward</th>
<th>improvement</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>W&amp;D</td>
<td>9.12</td>
<td>20.17%</td>
<td>0.000</td>
</tr>
<tr>
<td>DFM</td>
<td>9.23</td>
<td>18.75%</td>
<td>0.000</td>
</tr>
<tr>
<td>GRU</td>
<td>9.87</td>
<td>11.05%</td>
<td>0.000</td>
</tr>
<tr>
<td>HDQN</td>
<td>10.27</td>
<td>6.712%</td>
<td>0.002</td>
</tr>
<tr>
<td>DEAR</td>
<td>10.96</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Component Study

- DEAR-1: supervised training
- DEAR-2: no RNN
- DEAR-3: classical DQN (b)
- DEAR-4: no $Q(s, a) = V(s) + A(s, a)$
- DEAR-5: random ad
- DEAR-6: random location

Table 3: Component study results.

<table>
<thead>
<tr>
<th>variant</th>
<th>reward</th>
<th>improvement</th>
<th>$p$–value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAR-1</td>
<td>9.936</td>
<td>10.32%</td>
<td>0.000</td>
</tr>
<tr>
<td>DEAR-2</td>
<td>10.02</td>
<td>9.056%</td>
<td>0.000</td>
</tr>
<tr>
<td>DEAR-3</td>
<td>10.39</td>
<td>5.495%</td>
<td>0.001</td>
</tr>
<tr>
<td>DEAR-4</td>
<td>10.57</td>
<td>3.689%</td>
<td>0.006</td>
</tr>
<tr>
<td>DEAR-5</td>
<td>9.735</td>
<td>12.58%</td>
<td>0.000</td>
</tr>
<tr>
<td>DEAR-6</td>
<td>9.963</td>
<td>10.01%</td>
<td>0.000</td>
</tr>
<tr>
<td>DEAR</td>
<td>10.96</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Conclusion

- A deep RL framework DEAR with a novel DQN architecture for online advertising in recommender systems

- Determine three internally related actions at the same time
  - Interpolate an ad?
  - The optimal location?
  - The optimal ad?

- Simultaneously maximize the revenue of ads and minimize the negative influence of ads on user experience
Future Work

- Jointly optimizes advertising and recommending strategies
- More applications such as video games
Thanks

zhaoxi35@msu.edu

[QR Code Image]