Making Rewards More Rewarding: Sequential Learnable Environments for Deep Reinforcement Learning-based Sponsored Ranking

Nishan Subedi, AdKdd 2021

Chen Wang
Rutgers University
New Brunswick, United States
chen.wang.cs@rutgers.edu

Aidan Finn
Overstock.com
Midvale, United States
afinn@overstock.com

Nishan Subedi
Overstock.com
Midvale, United States
nsubedi@overstock.com
Background

- A good ranking function satisfies the need of all parties involved.
- CTR = click through rate
  \[ A_{\text{rank}} = bid \times CTR \]
- Dilemma of exploration makes Reinforcement Learning difficult
Simulated Environment Model in He et al. [AdKDD’18]

- **Ranking Problem formulation:**
  \[
  \phi(s, a, ad) = f_{a_1}(CTR) \cdot \text{bid} + a_2 \cdot f_{a_2}(CTR, CVR) + a_3 \cdot f_{a_3}(CVR, price)
  \]
  (CVR = Conversion rate = Purchase / View)

- **Click price:**
  \[
  \psi(ad, a) = a_2 \cdot f_{a_2}(CVR, CTR) + a_3 \cdot f_{a_3}(CVR, price);
  \]
  \[
  \text{click-price}_{\text{pred}} = \frac{\phi(ad', a) - \psi(ad, a)}{f_{a_1}(CTR)}.
  \]

- **Reward:**
  \[
  r = \text{click-price}_{\text{pred}} \cdot CTR.
  \]
Our Model: Improved Expressive Power with Simulated Environment Based On Sequential Trainable Embedding

- RNN-GRU Model where Ranking function depends on state ($s_t$) and advertisement ($z_t$) embeddings
  \[
  \hat{\phi}(ad, a) = \text{bid} \cdot f_{a_1}(\text{CTR}, s_t, z_t) + a_2 \cdot f_{a_3}(\text{CVR}, \text{CTR}, s_t, z_t) + a_4 \cdot f_{a_5}(\text{CVR}, \text{price}, s_t, z_t),
  \]
- The RNN is trainable -> able to improve reward prediction

\[
\mathcal{L}_{RNN} = ||\text{click-price}_{pred} - \text{click-price}||_2^2.
\]

(Similar idea explored by Wu et al., [Arxiv’18])

Figure 1: The framework of the trainable environment.
Our Model: Simulated Environment Loss Function

\[
\hat{\psi}(ad, a) = a_2 \cdot f_{a_3}(CVR, CTR, s_t, z_t) + a_4 \cdot f_{a_5}(CVR, price, s_t, z_t) \\
\phi(ad, a) = bid \cdot f_{a_1}(CTR, s_t, z_t) + \hat{\psi}(ad, a) \\
\text{price} = \frac{\hat{\phi}(ad', a) - \hat{\psi}(ad, a)}{f_{a_1}(CTR, s_t, z_t)} \\
\text{click-price}_{pred} = \text{sigmoid}(s_t^T w_{\text{scale}}) \cdot \text{price} \\
\mathcal{L}_{RNN} = \|\text{click-price}_{pred} - \text{click-price}\|_2^2.
\]

User & Advertiser Components

Ranking Strategy

Second Price Auction

Click price scaling

Loss function
Treatment for consistency in POSITIVE signals

- Clip click price prediction for positive samples to always be > 0

\[ \hat{\phi}(ad, a) = bid \cdot f_{a_1}(CTR, s_t, z_t) + a_2 \cdot f_{a_3}(CVR, CTR, s_t, z_t) + a_4 \cdot f_{a_5}(CVR, price, s_t, z_t), \]

- Action values (vector \(a\)) are encouraged to always increase for positive samples

\[ r = \text{click-price}_{pred} \cdot CTR + \lambda \frac{\|a\|}{\|a^{max}\|}, \]
Treatment for consistency in NEGATIVE signals

- Original click price allows increasing reward for negative values by increasing $a_1$

$$
\psi(ad,a) = a_2 \cdot f_{a_3}(CVR, CTR) + a_4 \cdot f_{a_5}(CVR, price);
$$

$$
\text{click-price}_{pred} = \frac{\phi(ad', a) - \psi(ad, a)}{f_{a_1}(CTR)}.
$$

- By adding a normalization term, decreasing $a_1$ increases the reward when raw-click-price < 0:

$$
\text{raw-click-price} = \phi(ad', a) - (a_2f_{a_3}(CVR, CTR, s_t, z_t) + a_4f_{a_5}(CVR, price, s_t, z_t))
$$

$$
\text{NM}(a_1) = \frac{f_{a_1}(CTR, s_t, z_t)}{f_{a_1}^{\min}(CTR, s_t, z_t)}
$$

$$
\text{click-price}_{pred} = \text{sigmoid}(s^T_t w_{scale}) \cdot \frac{(\text{raw-click-price} - \text{NM}(a_1))}{f_{a_1}^{\max}(CTR, s_t, z_t)}
$$
Agent: DDPG Continuous Policy Gradient

1. DDPG agent with a 25-layer policy network and 2-layer value network.
   a. Standard DDPG, simpler compared to that of He et al. [AdKDD’18]
   b. No heuristics driven parameter values -- easier to test across domains
   c. No online learning module - He et al. [AdKDD’18] acknowledges that lack of session information is the biggest reason to implement online update

Fair comparison: changes to the reward and the environment are agent-agnostic
Ranking Performance - training simulated env affects final score

Table 1: Ranking performance for different methods.

<table>
<thead>
<tr>
<th>Ranking Method</th>
<th>NDCG</th>
<th>Click</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid</td>
<td>0.0458</td>
<td></td>
</tr>
<tr>
<td>Purchase Rate</td>
<td>0.0485</td>
<td></td>
</tr>
<tr>
<td>AdRank</td>
<td>0.0530</td>
<td></td>
</tr>
<tr>
<td>CTR</td>
<td>0.0534</td>
<td></td>
</tr>
<tr>
<td>RL + original env ([4])</td>
<td>0.0530</td>
<td></td>
</tr>
<tr>
<td>RL + random new env (no training)</td>
<td>0.0529</td>
<td></td>
</tr>
<tr>
<td>RL + new env</td>
<td>0.0558</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Ranking performance for the environment model trained at different epochs. The most relevant set of features are used.

<table>
<thead>
<tr>
<th>Training Epochs</th>
<th>NDCG Click</th>
<th>percentage change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0529</td>
<td>–</td>
</tr>
<tr>
<td>1</td>
<td>0.0523</td>
<td>-1.13%</td>
</tr>
<tr>
<td>2</td>
<td>0.0530</td>
<td>+0.19%</td>
</tr>
<tr>
<td>5</td>
<td>0.0531</td>
<td>+0.38%</td>
</tr>
<tr>
<td>9</td>
<td>0.0536</td>
<td>+1.32%</td>
</tr>
<tr>
<td>14</td>
<td>0.0541</td>
<td>+2.27%</td>
</tr>
<tr>
<td>15</td>
<td>0.0558</td>
<td>+5.48%</td>
</tr>
</tbody>
</table>
Higher and more consistent rewards
Future steps:

Production deployment

Full scale for the agent for He et al. [AdKdd’18]

References:

Thankyou!
Overstock.com is Hiring!

Please reach out with questions/comments/feedback

Nishan Subedi
VP, Algorithms & ML
nsubedi@overstock.com
nishansubedi@gmail.com