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

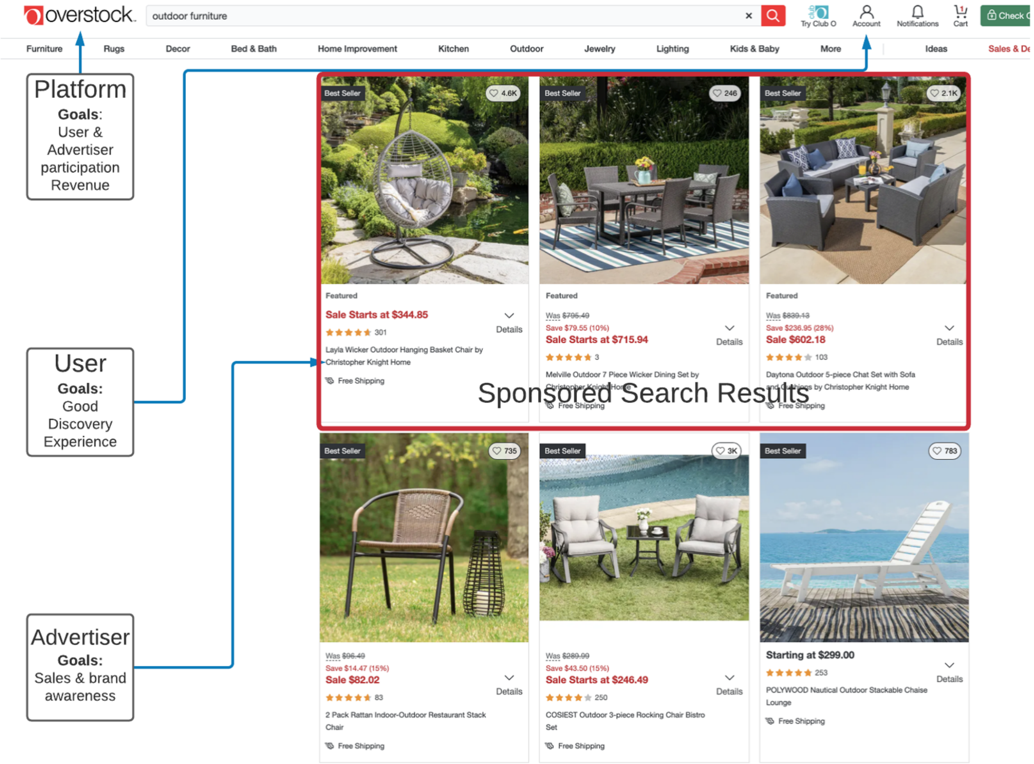
Nishan Subedi
Overstock.com
Midvale, United States
nsubedi@overstock.com

Background

- A good ranking function satisfies the need of all parties involved.
- CTR = click through rate

$$Adrank = bid * CTR$$

- Dilemma of exploration makes Reinforcement Learning difficult



Simulated Environment Model in He et al. [AdKDD'18]

- Ranking Problem formulation:

$$\phi(s, a, ad) = \underbrace{f_{a_1}(CTR) \cdot bid}_{\text{platform}} + \underbrace{a_2 \cdot f_{a_3}(CTR, CVR)}_{\text{user}} + \underbrace{a_4 \cdot f_{a_5}(CVR, price)}_{\text{advertiser}}$$

(CVR = Conversion rate = Purchase / View)

- Click price:

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

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

- Reward:

$$r = \text{click-price}_{pred} \cdot CTR.$$

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

$$\hat{\phi}(ad, \boldsymbol{a}) = bid \cdot f_{a_1}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) + a_2 \cdot f_{a_3}(CVR, CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) + a_4 \cdot f_{a_5}(CVR, price, \boldsymbol{s}_t, \boldsymbol{z}_t),$$

- 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])

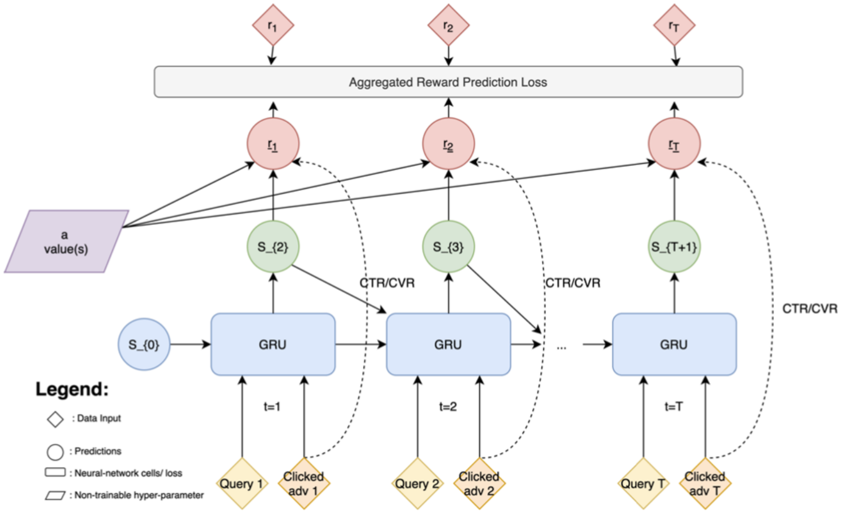


Figure 1: The framework of the trainable environment.

Our Model: Simulated Environment Loss Function

$$\hat{\psi}(ad, \boldsymbol{a}) = a_2 \cdot f_{a_3}(CVR, CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) + a_4 \cdot f_{a_5}(CVR, price, \boldsymbol{s}_t, \boldsymbol{z}_t)$$ **User & Advertiser Components**

$$\phi(ad, \boldsymbol{a}) = bid \cdot f_{a_1}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) + \hat{\psi}(ad, \boldsymbol{a})$$ **Ranking Strategy**

$$\text{price} = \frac{\hat{\phi}(ad', \boldsymbol{a}) - \hat{\psi}(ad, \boldsymbol{a})}{f_{a_1}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t)}$$ **Second Price Auction**

$$\text{click-price}_{pred} = \texttt{sigmoid}(\boldsymbol{s}_t^T \boldsymbol{w}_{\text{scale}}) \cdot \overline{\text{price}}$$ **Click price scaling**

$$\mathcal{L}_{RNN} = \|\text{click-price}_{pred} - \text{click-price}\|_2^2.$$ **Loss function**

Treatment for consistency in POSITIVE signals

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

$$\begin{aligned}\hat{\phi}(ad, \boldsymbol{a}) =& bid \cdot f_{a_1}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) + a_2 \cdot f_{a_3}(CVR, CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) \\ &+ a_4 \cdot f_{a_5}(CVR, price, \boldsymbol{s}_t, \boldsymbol{z}_t),\end{aligned}$$

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

$$r = \text{click-price}_{pred} \cdot CTR + \lambda \frac{||\boldsymbol{a}||}{||\boldsymbol{a}^{max}||},$$

Treatment for consistency in NEGATIVE signals

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

$$\psi(ad, \boldsymbol{a}) = a_2 \cdot f_{a_3}(CVR, CTR) + a_4 \cdot f_{a_5}(CVR, price);$$
$$\text{click-price}_{pred} = \frac{\phi(ad', \boldsymbol{a}) - \psi(ad, \boldsymbol{a})}{f_{a_1}(CTR)}.$$

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

$$\texttt{raw-click-price} = \hat{\phi}(ad', \boldsymbol{a}) - \big(a_2 f_{a_3}(CVR, CTR, \boldsymbol{s}_t, \boldsymbol{z}_t) + a_4 f_{a_5}(CVR, price, \boldsymbol{s}_t), \boldsymbol{z}_t\big)$$

$$\text{NM}(a_1) = \frac{f_{a_1}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t)}{f_{a_1^{min}}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t)}$$
$$\text{click-price}_{pred} = \texttt{sigmoid}(\boldsymbol{s}_t^T \boldsymbol{w}_{\text{scale}}) \cdot \frac{\big(\texttt{raw-click-price} - \text{NM}(a_1)\big)}{f_{a_1^{max}}(CTR, \boldsymbol{s}_t, \boldsymbol{z}_t)},$$

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

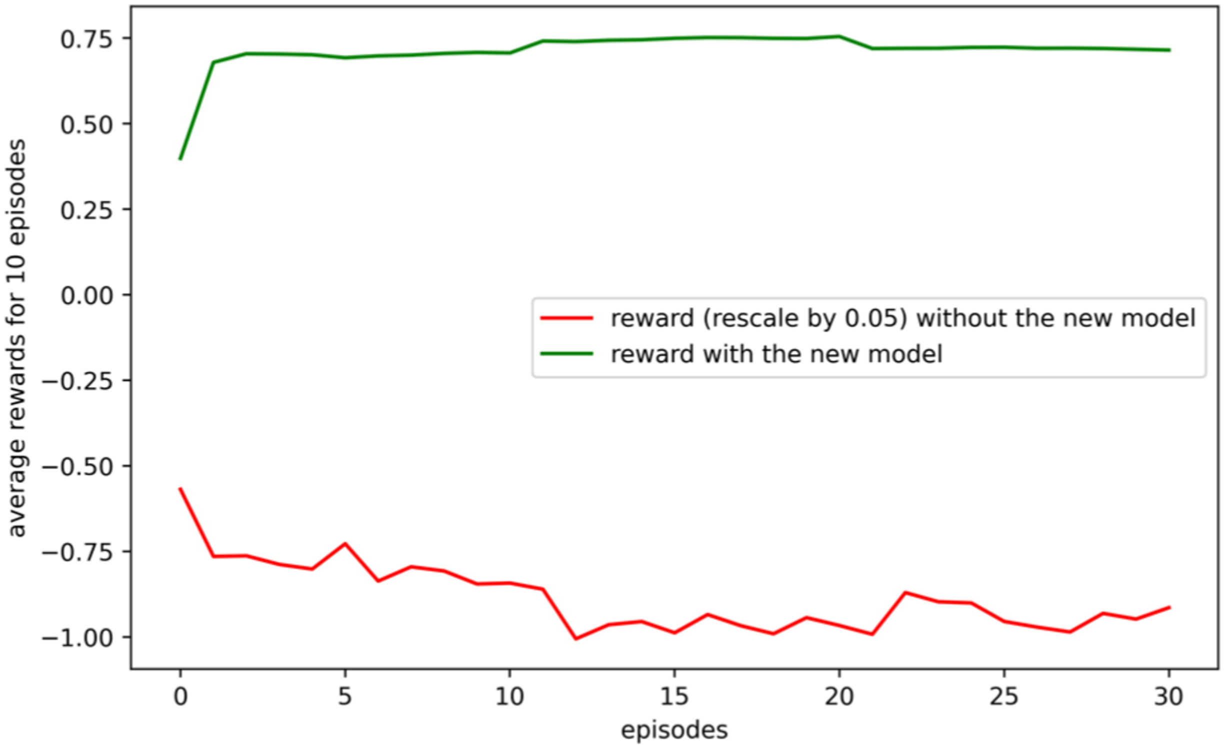
Table 1: Ranking performance for different methods.

Ranking Method	NDCG Click
Bid	0.0458
Purchase Rate	0.0485
AdRank	0.0530
CTR	0.0534
RL + original env ([4])	0.0530
RL + random new env (no training)	0.0529
RL + new env	**0.0558**

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

Training Epochs	NDCG Click	percentage change
0	0.0529	–
1	0.0523	-1.13%
2	0.0530	+0.19%
5	0.0531	+0.38%
9	0.0536	+1.32%
14	0.0541	+2.27%
15	0.0558	+5.48%

Higher and more consistent rewards



Future steps:

Production deployment

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

References:

- Li He, Liang Wang, Kaipeng Liu, and Weinan Zhang. 2018. Deep Policy Optimization for E-commerce Sponsored Search Ranking Strategy. In Proceedings of the 2018 AdKDD and TargetAd.
- Wenjin Wu, Guojun Liu, Hui Ye, Chenshuang Zhang, Tianshu Wu, Daorui Xiao, Wei Lin, and Xiaoyu Zhu. 2018. EENMF: An End-to-End Neural Matching Framework for E-Commerce Sponsored Search. CoRR abs/1812.01190 (2018). arXiv:1812.01190 http://arxiv.org/abs/1812.01190

Thankyou!
Overstock.com is Hiring!

Please reach out with questions/comments/feedback

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

overstock™