Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning

Xiangyu Zhao[1], Liang Zhang[2], Zhuoye Ding[2], Long Xia[2], Jiliang Tang[1] and Dawei Yin[2]
1: Data Science and Engineering Lab, Michigan State University 2: Data Science Lab, JD.com

Data Science and Engineering Lab

My homepage

JD.COM 京东

Motivation

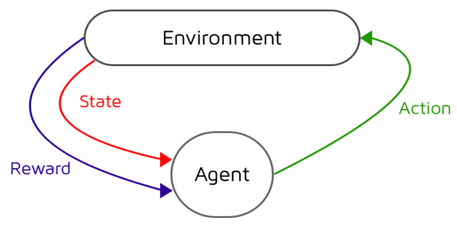
Recommender systems can mitigate the information overload problem by suggesting users' personalized items

❑ Challenges of Existing Recommender Systems

- ❖ Recommendation procedure as static process
- ❖ Making recommendations following fixed greedy strategy
- ❖ Maximizing the immediate (short-term) reward from users

❑ Why Reinforcement Learning?

- ❖ Continuously updating the recommendation strategies



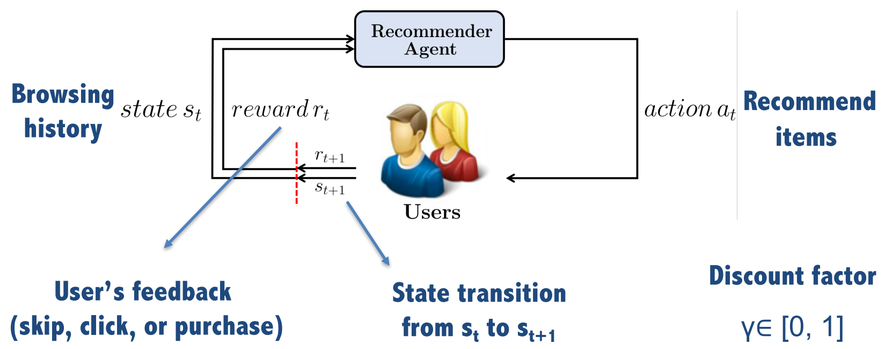
- ❖ Maximizing the long-term reward from users

❑ Why Negative Feedback?

- ❖ Positive: click or purchase
- ❖ Negative: skip
- ❖ What users may not like

Problem Statement

❑ Markov Decision Process (MDP)



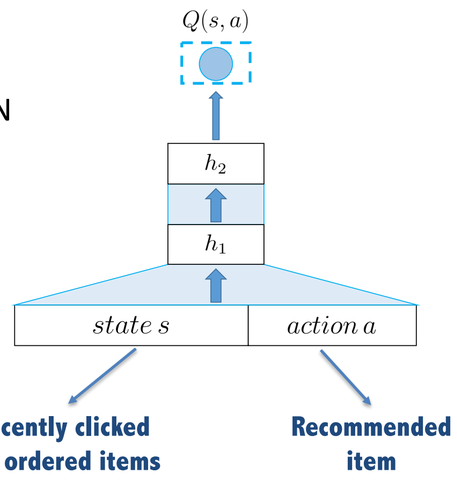
❑ **Goal:** find a recommendation policy π : S → A, which can maximize the cumulative reward for the recommender system

Basic DQN Model

❑ Only Positive Feedback

- ❖ State s = $\{i_1, \ldots, i_N\}$ is defined as the previous N items that a user clicked/purchased recently
- ❖ Transition from s to s':
  - ➢ If the user skips the item, then s' = s
  - ➢ If the user clicks/purchases the item, then s' = $\{i_2, \ldots, i_N, a\}$

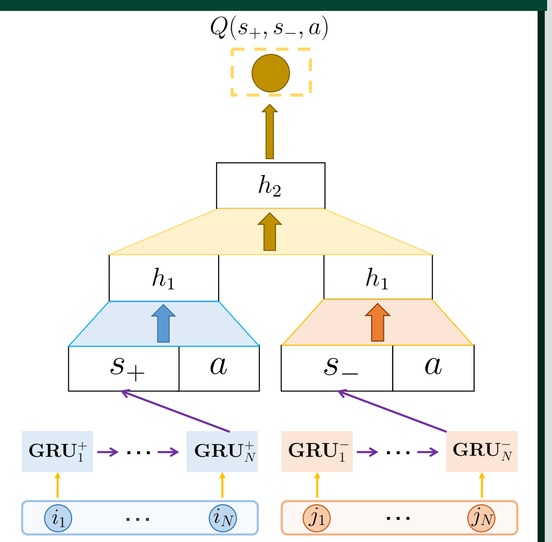
$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') | s, a\right]$$



The Proposed Framework

❑ The Architecture of DEERS

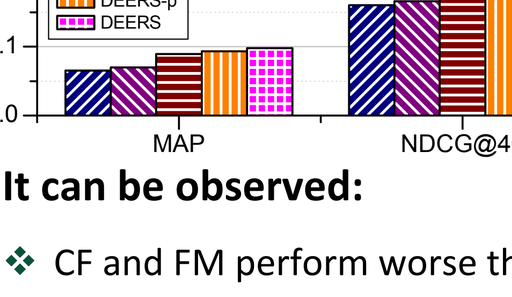
- ❖ State s = (s_+, s_-), where $s_+ = \{i_1, \ldots, i_N\}$ is the previous N clicked or purchased items, $s_- = \{j_1, \ldots, j_N\}$ is the previous N skipped items
- ❖ Transition from s to s':
  - ➢ If the user skips the item, then $s'_- = \{j_2, \ldots, j_N, a\}$
  - ➢ If the user clicks/purchases the item, then $s'_+ = \{i_2, \ldots, i_N, a\}$



- ❖ RNN with GRU to capture users' sequential preference
- ❖ Recommend an item that is similar to the clicked/purchased items (left part), while dissimilar to the skipped items (right part)

❑ The Pairwise Regularization Term

- ❖ RA often recommends items belong to the same category, while users click/purchase a part of them and skip others



Time	State	Item	Category	Feedback
1	s_1	a_1	A	skip
2	s_2	a_2	B	click
3	s_3	a_3	A	click
4	s_4	a_4	C	skip
5	s_5	a_5	B	skip
6	s_6	a_6	A	skip
7	s_7	a_7	C	order

$$L(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(y - Q(s_+, s_-, a; \theta)\right)^2 - \alpha\left(Q(s_+, s_-, a; \theta) - Q(s_+, s_-, a^E; \theta)\right)^2\right]$$

Experiment

- ❖ Dataset from JD.com
- ❖ Baselines
  - ➢ Collaborative filtering
  - ➢ Factorization Machines
  - ➢ RNN with GRU
  - ➢ DEERS-p(only positive feedback)

❑ It can be observed:

- ❖ CF and FM perform worse than GRU and DEERS, since CF and FM ignore the temporal sequence of the users' browsing history
- ❖ GRU performs worse than than DEERS-p, since GRU maximizes the immediate reward for recommendations
- ❖ DEERS performs better than DEERS-p because DEERS integratses both positive and negative items (or feedback)

Conclusion

- ❖ We design a novel architecture to capture both positive and negative feedback simultaneously
- ❖ We design a pairwise regularization term to maximize the difference of Q-values between competing items
- ❖ This work is supported by the National Science Foundation (NSF) under grant number IIS-1714741 and IIS-1715940