Reinforcement learning for long-term reward optimization in recommender systems

Anton DOROZHKO, Evgeniy PAVLOVSKIY

July 2, 2019

Outline

Recommendation task

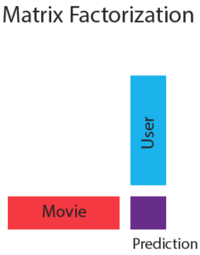
Observations

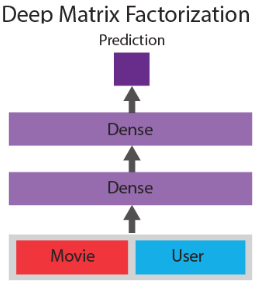
$(x, y, r(x, y)) \in \mathcal{X} \times \mathcal{Y} \times \mathbb{R}$

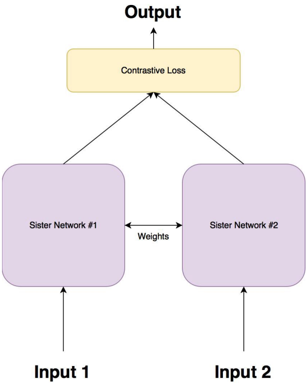
Recommender System

takes as input a sequence of observations and outputs a mapping:
Deterministic : $\mathcal{X} \to \mathcal{Y}$
Probabilistic : $\mathcal{X} \to \Delta(\mathcal{Y})$

Recommendations







Siamese network

Time information

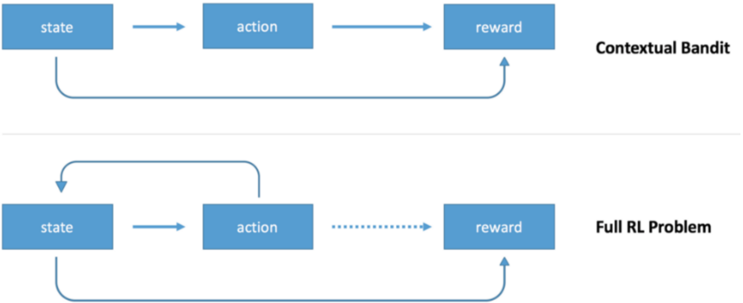
Sequence

	user_id	item_id	rating	timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291

Matrix

item_id	1	2	3	4	5	6	7
user_id							
1	5.0	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	2.0	NaN
6	4.0	NaN	NaN	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	4.0	NaN
8	4.0	NaN	NaN	3.0	NaN	NaN	NaN
9	5.0	NaN	NaN	NaN	NaN	NaN	NaN
10	5.0	5.0	NaN	NaN	NaN	NaN	4.0

RL vs Contextual Bandits

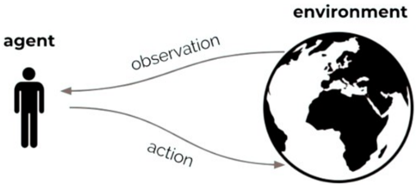


Assumption

States (observations) in Contextual Bandits are i.i.d

Why Reinforcement Learning

- users' preferences are dynamic
- optimization of the long-term objective
- entering the cycle of recommendations



Reinforcement Learning

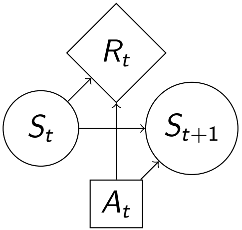
Learn to make a good sequences of decisions

Reinforcement learning

Markov Decision Process MDP

MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$

1. $\mathcal{S}$ - set of states
2. $\mathcal{A}$ - set of actions
3. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ - *transition function* $p(s_{t+1}|s_t, a_t)$
4. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$- rewards



Markov property assumption

$$p(r_t, s_{t+1}|s_0, a_0, r0, ..., s_t, a_t) = p(r_t, s_{t+1}|s_t, a_t)$$

Reinforcement learning

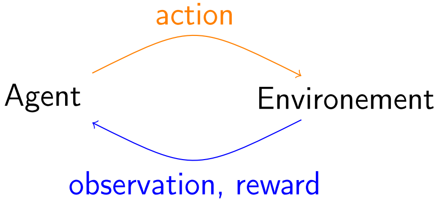
Discounted rewards

$$G_t = R_t + \gamma R_{t+1} \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$$\max_{\pi_\theta} \mathbb{E}_{\pi_\theta}[G_0]$$

$\pi_\theta : \mathcal{S} \to \mathcal{A}$ - agent policy

Interaction



Modelization

MDP $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$

- **State space** $s_t = \{s_t^1, ..., s_t^N\} \in \mathcal{S}$ - N last items
- **Action space** $a_t = \{a_t^1, ..., a_t^K\} \in \mathcal{A}$ - list of items
- **Reward** $\mathcal{R}$ $r(s_t, a_t) \in \mathbb{R}$
- **Transition function** $\mathcal{P}$

$$s_{t+1} = \begin{cases} \text{add } a_t^k \rightarrow s_{t+1}, \text{remove first } s_{t+1} & r_t > 0 \\ s_t & \textit{otherwise} \end{cases}$$

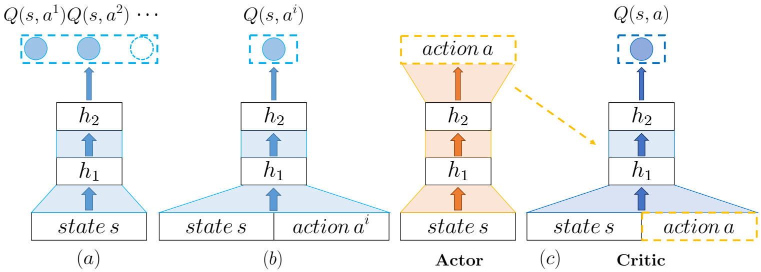
- **Discount** $\gamma \in [0, 1]$

Q-learning

Action-value function

The action-value function $Q_\pi(s, a)$ is expected return starting from state s, taking action a, and then following policy π

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$



Q-learning

Draw N transitions from Experience Replay

$$< s, a, r, s' >\sim D$$

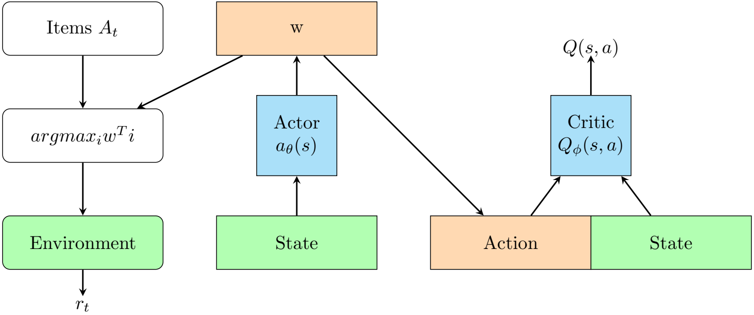
Update Critic network:

$$\nabla_\phi \frac{1}{N} \sum_j (r + \gamma Q_\phi(s', a_\theta(s')) - Q_\phi(s, a))^2$$

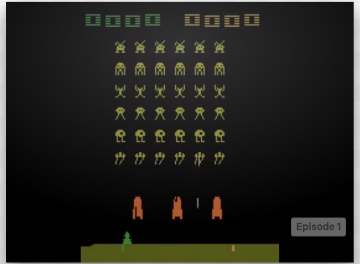
Update Actor network:

$$\nabla_\theta \frac{1}{N} \sum_j Q_\phi(s, a_\theta(s))$$

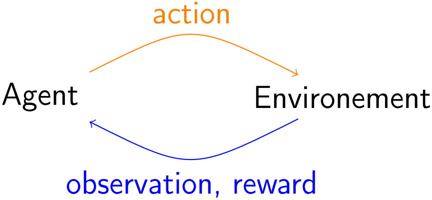
Proto action idea



Environnement with OpenAI Gym [1]



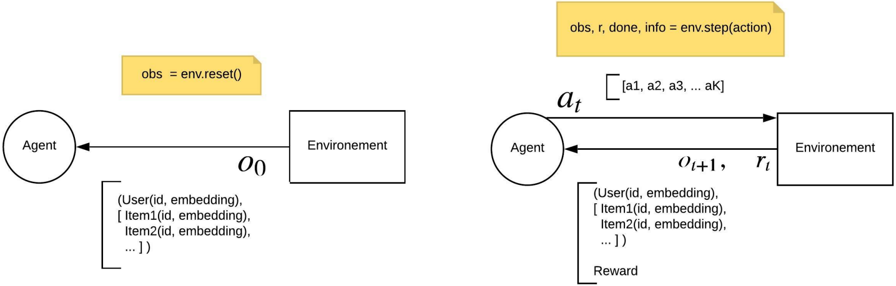
RandomAgent on SpaceInvaders-v0




```
import gym
env = gym.make("Taxi-v1")
observation = env.reset()
for _ in range(1000):
  env.render()
  action = env.action_space.sample() # your agent here (this takes random act
  observation, reward, done, info = env.step(action)
```

[1] https://gym.openai.com/

Base Interface of recommendation env



Parameters study

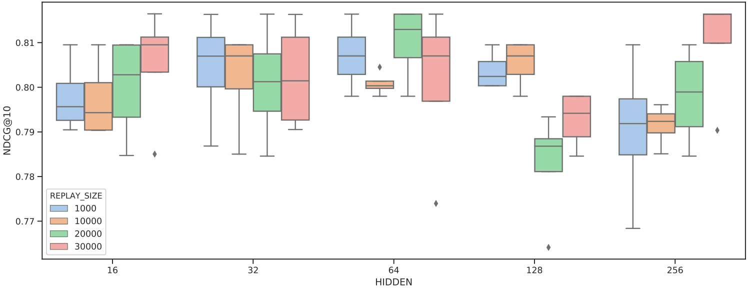


Figure: Parameter study of DDPG algorithm on MovieLens100k

Offline evaluation

Table: Metrics for different session size for MovieLens 1M

	ml-1m s20		**ml-1m s30**	
	NDCG@10	Precision@10	NDCG@10	Precision@10
Random	0.76	0.65	0.78	0.66
Popularity	0.85	0.78	**0.88**	**0.81**
SVD	0.76	0.66	0.77	0.66
LinUCB	0.85	0.78	0.87	0.8
DDPG	**0.87**	**0.81**	0.83	0.74

MovieLens-1M

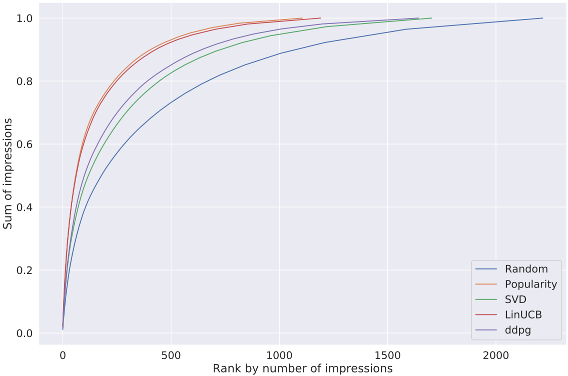


Figure: Distribution of impressions over items for MovieLens 1M: lower curve means that algorithm served more different items

Results

The main contributions of the project are 2 folds:

1. Propose and build recommendation environments/benchmarks with OpenAI Gym interface
2. Parameter study for DDPG agent

Conclusion

That research direction is very promising and vast. Main advantages of the application of RL to recommendation process are

- we consider recommendation process as dynamic and optimize for long-term rewards
- we model the influence of the recommendations on user state
- MDP formalism is flexible and different scenarios can be modeled easily in this framework

Proper evaluation is hard *Rendle S., Zhang L., Koren Y.* On the Difficulty of Evaluating Baselines: A Study on Recommender Systems. // CoRR. 2019. Vol. abs/1905.01395

Reinforcement learning for long-term reward optimization in recommender systems

Anton DOROZHKO, Evgeniy PAVLOVSKIY

July 2, 2019

References I

- An Introduction to Deep Reinforcement Learning. /. — V. Francois-Lavet [et al.]. — 2018. — arXiv: 1811.12560 [cs.LG].
- *Bonner S.*, *Vasile F.* — Causal Embeddings for Recommendation. — 2017. — June. — DOI: 10.1145/3240323.3240360. — arXiv: 1706.07639. — URL: http://arxiv.org/abs/1706.07639.
- Deep Reinforcement Learning for Page-wise Recommendations. /. — X. Zhao [et al.]. — 2018. — May. — arXiv: 1805.02343. — URL: http://arxiv.org/abs/1805.02343.

References II

- E-commerce in Your Inbox. /. — M. Grbovic [et al.] // Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15. — 2015. — URL: http://dx.doi.org/10.1145/2783258.2788627.
- *Liebman E.*, *Saar-Tsechansky M.*, *Stone P.* — DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation. — 2014. — Jan. — arXiv: 1401.1880. — URL: http://arxiv.org/abs/1401.1880.
- RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. /. — D. Rohde [et al.]. — 2018. — Aug. — URL: http://arxiv.org/abs/1808.00720.

References III

Recommendations with Negative Feedback via Pairwise Deep Reinforcement. /. — X. Zhao [et al.] // KDD. — 2018. — Vol. 18. — arXiv: 1802.06501v3.

Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. /. — X. Zhao [et al.] // KDD. — 2018. — arXiv: 1802.06501v3.

Rendle S., *Zhang L.*, *Koren Y.* — On the Difficulty of Evaluating Baselines: A Study on Recommender Systems. — // CoRR. — 2019. — Vol. abs/1905.01395. — arXiv: 1905.01395. — URL: http://arxiv.org/abs/1905.01395.

References IV

Returning is Believing. /. — Q. Wu [et al.] // Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17. — New York, New York, USA : ACM Press, 2017. — P. 1927–1936. — ISBN 9781450349185.

Stabilizing Reinforcement Learning in Dynamic Environment with Application to Online Recommendation. /. — S.-Y. Chen [et al.] // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18. — New York, New York, USA : ACM Press, 2018. — P. 1187–1196. — ISBN 9781450355520. — DOI: 10.1145/3219819.3220122.