

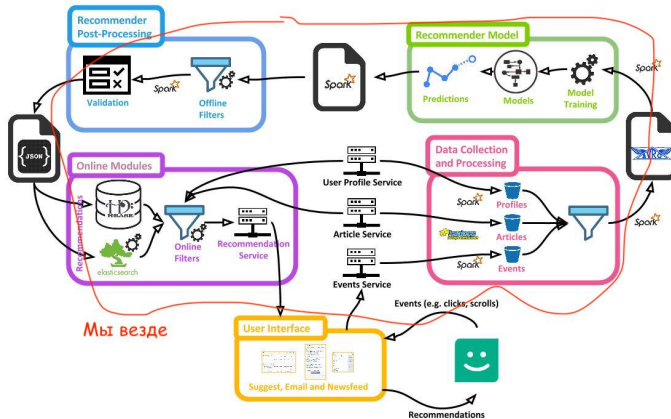
Recommendations + Reinforcement Learning = ♥

Николай Анохин

16 ноября 2023 г.



## Контекст



## Сложности в постановке задачи рекомендаций

1. Оцениваем айтемы по-отдельности, а показываем по несколько (лентой)
2. Модель не объясняет, почему именно эти айтемы подходят пользователю
3. Смещение между распределениями на обучении и применении
4. Не учитывается долгосрочный эффект рекомендаций



## Долгосрочный эффект рекомендаций



Долгосрочный эффект рекомендаций  
○○●○○○○

Многорукие бандиты  
○○○○○○○

Симуляторы для рекомендаций  
○○

Полная постановка RL в рекомендациях  
○○○

Итоги  
○○○○○

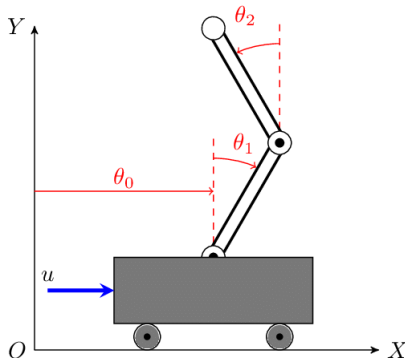
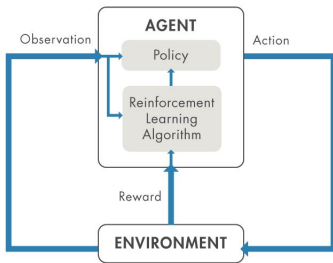


## Долгосрочный эффект рекомендаций

1. Эволюция пользователя (рекомендер влияет на пользователя)
2. Эволюция рекомендера (рекомендер влияет на себя)
3. Отложенная награда



## Постановка задачи Reinforcement Learning



## Markov Decision Process (MDP)

История	$H_t = O_1, A_1, R_1, \dots O_t, A_t, R_t$
Состояние	$S_t = f(H_t)$
Среда	$\mathcal{P}(S_t A_t, S_{t-1})$
Награда	$R(S_t S_{t-1})$
Политика	$\pi(A S)$
Кумулятивная награда	$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots$

Цель: выбрать оптимальную политику

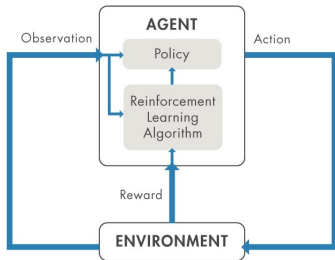
MDP:  $(S, A, \mathcal{P}, R)$

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\mathcal{P}, \pi} G_t$$





## Рекомендации как Reinforcement Learning



RecSys

Пользователь

Контекст

Рекомендательный сервис

Алгоритм рекомендаций

Рекомендация

Покупка, просмотр, клик

???

→

RL

→

Среда (environment)

→

Наблюдение (observation)

→

Агент (agent)

→

Политика (policy)

→

Действие (action)

→

Награда (reward)

→

Эпизод (episode)



## Почему RL (почти) не используется в продакшен рекомендерах?

- Огромное меняющееся пространство действий-состояний
- Отсутствие данных (сред) для проверки идей
- Дорогая реализация алгоритмов



Долгосрочный эффект рекомендаций  
○○○○○○○

**Многорукие бандиты**  
●○○○○○○

Симуляторы для рекомендаций  
○○

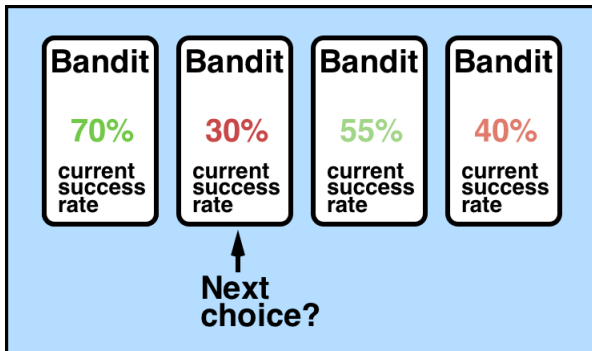
Полная постановка RL в рекомендациях  
○○○

Итоги  
○○○○○

## Многорукие бандиты



## Multi-armed bandit



$$Q_n(a) = \mathbb{E}[R_n \mid A_n = a]$$

$$A_n^* = \max_a Q_n(a)$$



## Варианты решений I [BAN19b]

- $\varepsilon$ -greedy: выбираем случайную руку с вероятностью  $\varepsilon$ , иначе жадно
- $\varepsilon$ -decay: как  $\varepsilon$ -greedy, но уменьшаем  $\varepsilon$  со временем

$$\varepsilon(n) = \frac{1}{1 + n\beta}$$

- Upper Confidence Bound (UCB)

$$A_n = \arg \max_a \left( Q_n(a) + c \sqrt{\frac{\log(n)}{N_n(a)}} \right)$$



## Варианты решений II: Gradient Bandit [BAN19c]

Политика, которая чаще выбирает “хорошие” руки

$H(A_k)$  – value руки  $k$

$$\pi(A_k) = \frac{\exp H(A_k)}{\sum_j \exp H(A_j)}$$

Обновление

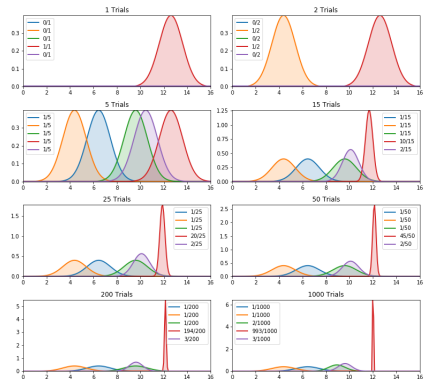
$$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t))$$

$$H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a), \forall a \neq A_t$$

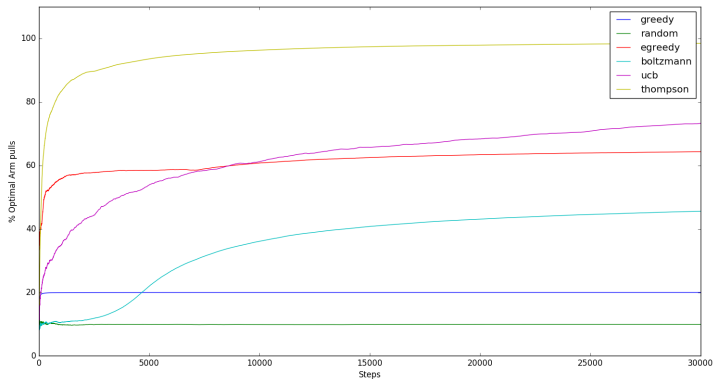


## Варианты решений III: Thompson Sampling

1. Для каждой руки оцениваем распределение награды
2. Семплируем значение из каждого из распределений
3. Выбираем руку с наибольшим значением



## Сравнение алгоритмов<sup>1</sup>



<sup>1</sup><https://sudeeppraja.github.io/Bandits/>





## Итоги

- (В некоторых случаях) оптимально соблюдают баланс Explore/Exploit
- Простые и работают на практике для задач с небольшим количеством действий

- Не учитывают состояния среды



Долгосрочный эффект рекомендаций  
○○○○○○○

Многорукие бандиты  
○○○○○○○

Симуляторы для рекомендаций  
●○

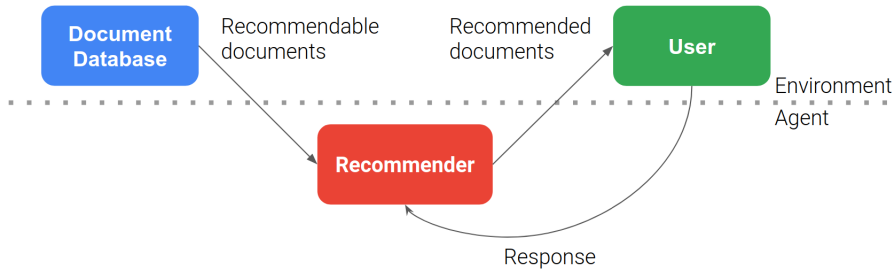
Полная постановка RL в рекомендациях  
○○○

Итоги  
○○○○○

## Симуляторы для рекомендаций



# RecSim: A Configurable Simulation Platform for Recommender Systems [IHM<sup>+</sup>19]



Долгосрочный эффект рекомендаций  
○○○○○○○

Многорукие бандиты  
○○○○○○○

Симуляторы для рекомендаций  
○○

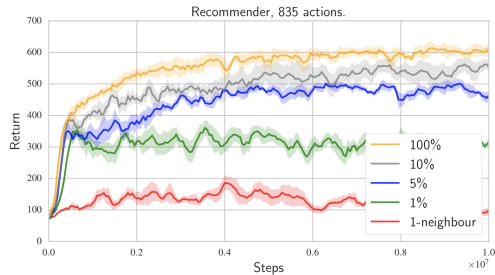
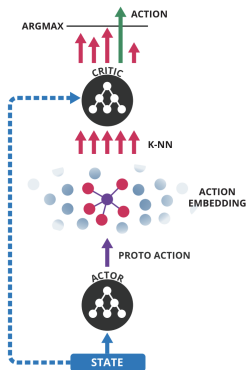
Полная постановка RL в рекомендациях  
●○○

Итоги  
○○○○○

## Полная постановка RL в рекомендациях



# Deep Reinforcement Learning in Large Discrete Action Spaces [DAEH<sup>+</sup>15]<sup>2</sup>

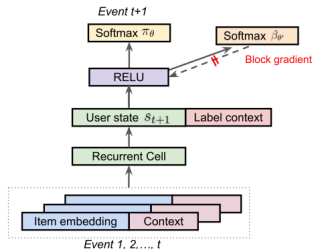


<sup>2</sup>Пример использования в рекомендациях: <https://arxiv.org/abs/1811.05869>



# Top-K Off-Policy Correction for a REINFORCE Recommender System [CBC<sup>+</sup>18]

- Масштабировали алгоритм REINFORCE на огромное пространство действий.
- Применили корректировку смещения между logging и обучаемой политикой.
- Изобрели новую корректировку на top-k рекомендации.
- Применили все это в продакшене YouTube.



Долгосрочный эффект рекомендаций  
○○○○○○○

Многорукие бандиты  
○○○○○○○

Симуляторы для рекомендаций  
○○

Полная постановка RL в рекомендациях  
○○○

Итоги  
●○○○○○

Итоги



## Итоги

Постановка задачи RL очень хорошо соответствует задаче рекомендаций.

В рекомендациях все признают проблемы explore/exploit и смещений. Их решают методами, заимствованными из RL.

Придется подождать, пока RL в рекомендациях станет общей практикой.



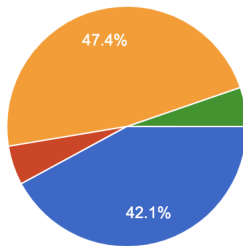


## Итоги курса

В будущем рекомендательные системы будут давать релевантные, разнообразные и полезные рекомендации. Они будут учитывать долгосрочные интересы пользователей. А пользователи будут понимать, почему им что-то предлагают и смогут контролировать механизмы построения рекомендаций.

Но понадобится ваша помощь. И научная честность.





- Хочу получить базовое введение в рекомендательные сервисы
- Хочу получить глубокие теоретические знания о задаче рекомендаций
- Хочу научиться создавать боевые рекомендательные сервисы
- Хочу просто пройти курс, это требуется, чтобы закончить МФТИ



Долгосрочный эффект рекомендаций  
○○○○○○○

Многорукие бандиты  
○○○○○○○

Симуляторы для рекомендаций  
○○

Полная постановка RL в рекомендациях  
○○○






Итоги  
○○○○●



<https://t.me/mlvok>




## Литература I

-  *13 solutions to multi-arm bandit problem for non-mathematicians*, 2019.
-  *Multi-armed bandits and reinforcement learning*, 2019.
-  *Multi-armed bandits and reinforcement learning 2*, 2019.
-  Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi, *Top-k off-policy correction for a REINFORCE recommender system*, CoRR **abs/1812.02353** (2018).
-  Gabriel Dulac-Arnold, Richard Evans, H. V. Hasselt, Peter Sunehag, Timothy P. Lillicrap, Jonathan J. Hunt, Timothy A. Mann, Théophane Weber, Thomas Degris, and Ben Coppin, *Deep reinforcement learning in large discrete action spaces*, arXiv: Artificial Intelligence (2015).



## Литература II

 Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier, *Recsim: A configurable simulation platform for recommender systems*, 2019, cite arxiv:1909.04847.

