

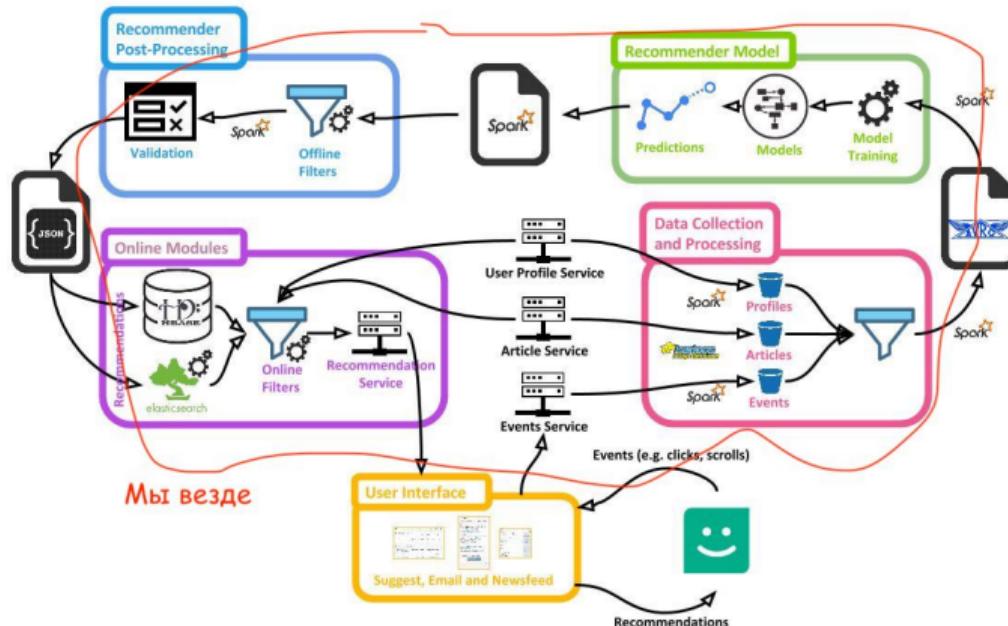
Нерешенные проблемы и новые направления

Николай Анохин

23 июня 2022 г.



Контекст



Что мы уже умеем

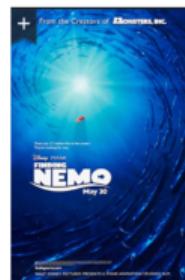
$$\hat{r}_{ui} = f_{\theta}(x_u, x_i, x_c)$$

Проблемы

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 долгострочный эффект рекомендаций

Разнообразие в рекомендательных системах

Разнообразие / Diversity



Набираем айтемы с разными аспектами

f - аспект (признак) айтема, $p(f|i)$ – вероятность найти аспект у айтема i

Распределение аспекта у пользователя

$$p(f|u) = \frac{\sum_{i \in I_u} p(f|i)}{|I_u|}$$

Распределение аспекта в рекомендациях

$$q(f|u) = \frac{\sum_{i \in RL} p(f|i)}{|RL|}$$

Формируем список так, чтобы $q(f|u)$ совпало с $p(f|u)$

Жадное переранжирование

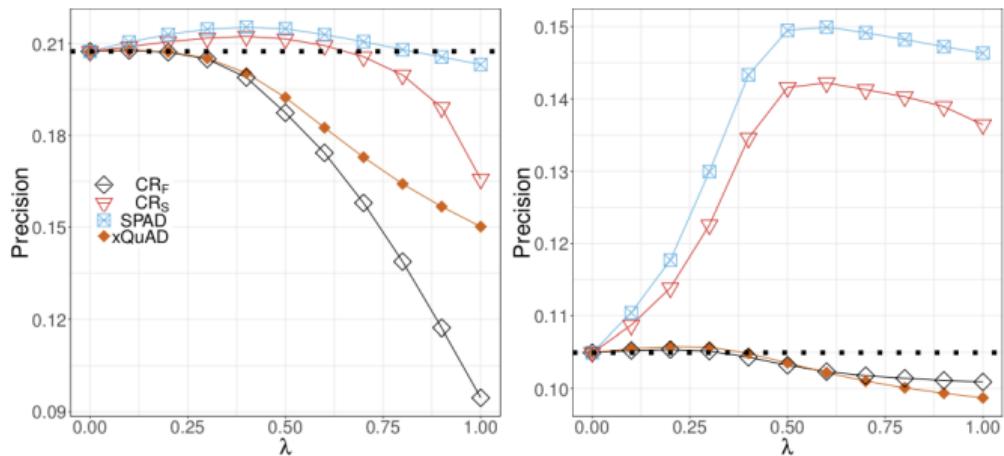
Добавляем в список рекомендаций айтем с максимальным значением

$$(1 - \lambda) \cdot s(u, i) + \lambda \cdot gain(i, RL),$$

пока не получим список нужной длины.

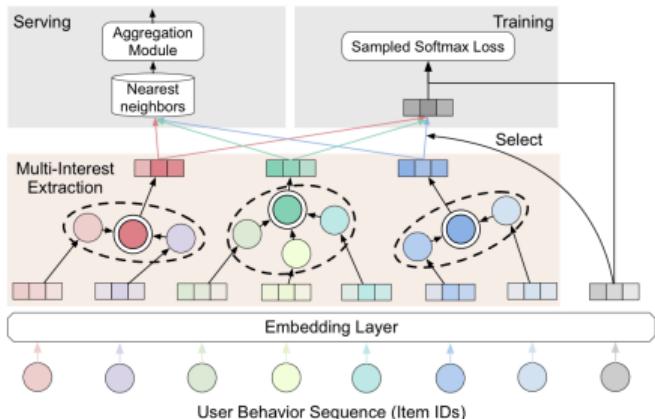
- $s(u, i)$ – релевантность айтема i для пользователя u
- $gain(i, RL) = div(RL \cup \{i\}) - div(RL)$ – улучшение разнообразия при добавлении айтема
- λ – гиперпараметр

A Comparison of Calibrated and Intent-Aware Recommendations [KB19]



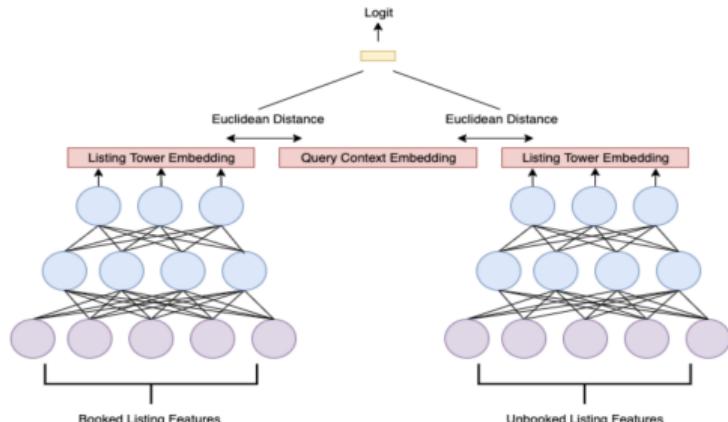
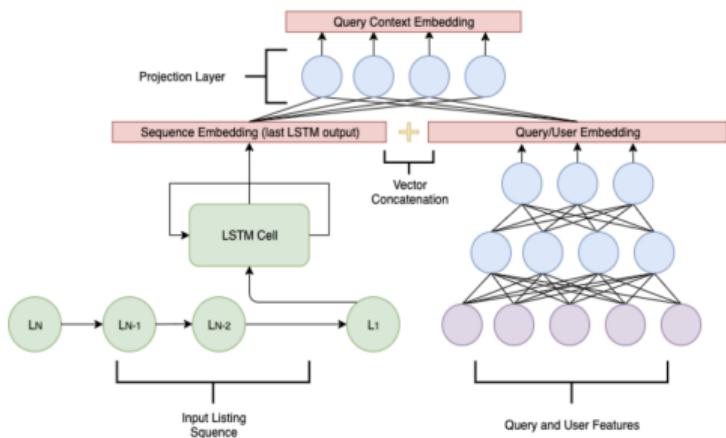
Учим разнообразие вместе с моделью 1

Controllable Multi-Interest Framework for Recommendation [CZZ⁺20]



Учим разнообразие вместе с моделью 2

Managing Diversity in Airbnb Search [AHR⁺20]



Из-за поточечного предсказания релевантности, приходится дополнительно разнообразить списки рекомендаций

Необходимость разнообразия обосновывается A/B экспериментом

Разнообразие в рекомендательных системах
ooooooooo

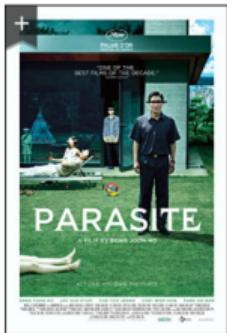
Объяснение рекомендаций
●oooooooooo

Смещения Долгосрочный эффект рекомендаций Многорукие банды
oooooooooooo ooooo

Объяснение рекомендаций

Объяснения

???

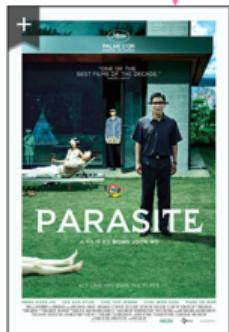


Объяснения

Потому что
вы оценили
Toy Story



Потому что вам
нравятся комедии



Популярное



Зачем объяснять рекомендации?

1. Прозрачность: объяснить пользователю, как работает система
2. Контролируемость: позволить пользователю исправить ошибки
3. Доверие: убедить пользователя, что система работает правильно
4. Убеждение: мотивировать пользователя к покупке
5. Полезность: помочь пользователю принять правильное решение
6. Эффективность: помочь пользователю принять решение быстро
7. Удовольствие: сделать приятно пользователю

Case-based

Because you have selected or highly rated: Movie A

Потому что вы смотрели «ЛЕГО Фильм 2»



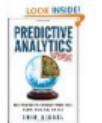
Collaborative

Customers Who Bought This Item Also Bought A

Customers Who Bought This Item Also Bought



Predictive Analytics For Dummies
› Anasse Bari
★★★★★ 29
Paperback
\$17.72



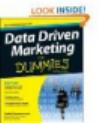
Predictive Analytics: The Power to Predict Who...
› Eric Siegel
★★★★★ 229
#1 Best Seller in Econometrics
Hardcover
\$16.88



Quantifying the User Experience: Practical...
› Jeff Sauro
★★★★★ 8
Paperback
\$40.63



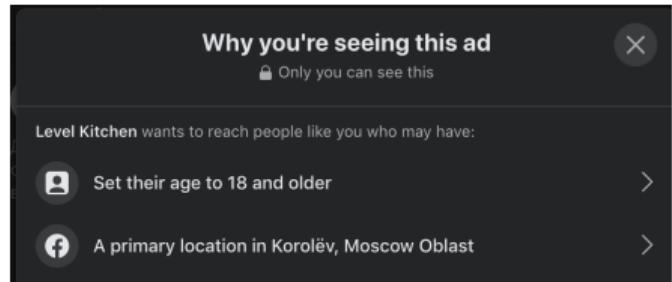
Marketing Analytics: Strategic Models and...
› Stephan Sorger
★★★★★ 29
Paperback
\$50.52



Data Driven Marketing For Dummies
› David Semmelroth
Paperback
\$20.49

Content-based

Recommended because you said liked science fiction



Knowledge-based

Less Memory and Lower Resolution and Cheaper



This item **Lenovo IdeaPad 3 14"** Laptop, Intel Core i3-1005G1 Processor, 4GB DDR4 RAM, 128GB M.2 SSD Storage, 14.0" FHD (1920 x 1080) Display, Integrated Graphics, Windows 10 S, 81WDD010QUS, Platinum Grey

#1 Best Seller

Add to Cart



Lenovo IdeaPad 3 14 Laptop, AMD Ryzen 5 5500U Processor, 8GB DDR4 RAM, 256GB NVMe SSD Storage, 14.0" FHD (1920 x 1080) Display, AMD Radeon 7 Graphics, Windows 10 Home, 82KTD00AMUS, Abyss Blue

Add to Cart



Lenovo IdeaPad 1 14 14.0" Laptop, 14.0" HD (1366 x 768) Display, Intel Celeron N4020 Processor, 4GB DDR4 RAM, 64 GB SSD Storage, Intel UHD Graphics 600, Win 10 in S Mode, 81VU0079US, Ice Blue

Add to Cart



Lenovo IdeaPad Gaming 3 15 15.6" Laptop, 15.6" FHD (1920 x 1080) Display, AMD Ryzen 5 5600H Processor, NVIDIA GeForce GTX 1650, 8GB DDR4 RAM, 256GB SSD Storage, Windows 10H, 82K20015US, Shadow Black

Add to Cart

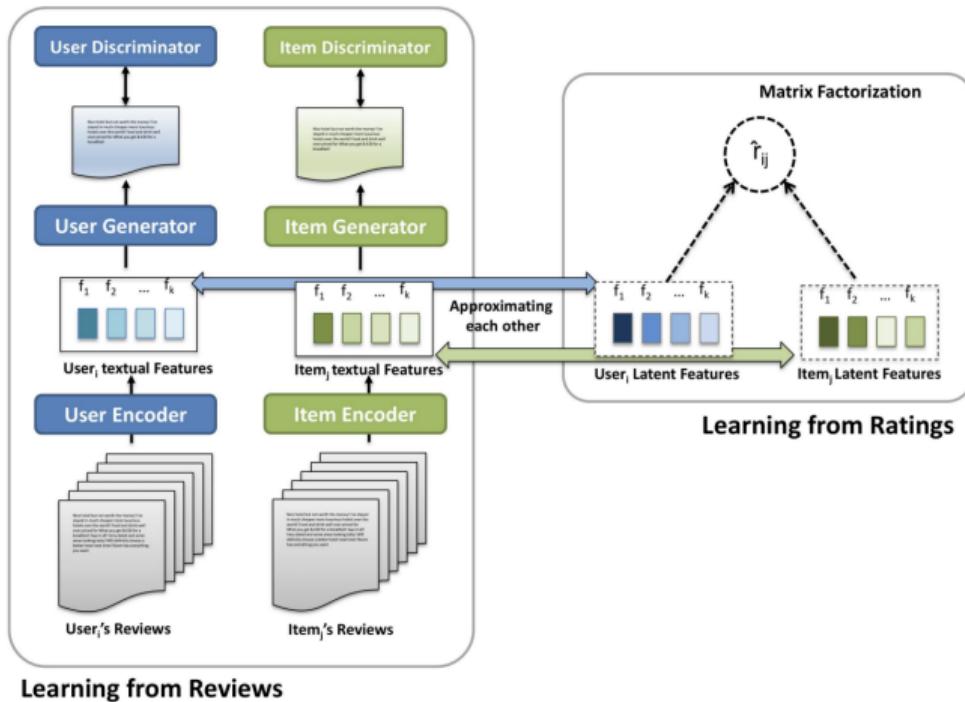
Customer Rating	★★★★☆ (32)	★★★★★ (118)	★★★★★ (590)	★★★★★ (134)
Price	\$399 ⁰⁰	\$539 ⁰⁰	\$266 ⁴²	\$769 ²²
Sold By	eSales Plus	Xocean	Mohawk Shop	ETRON INC - ELECTRONICS SUPPLIER
Computer Memory Size	128 GB	8	4	8
CPU Model Manufacturer	Intel	AMD	Intel	AMD
CPU Speed	1.2 GHz	2.1	1.1	3.3

Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits [MLH⁺18]

Explanation	# Impressions
Because it's [day of week]	140.3K
Inspired by [user]'s recent listening	138.4K
Because it's a new release	140.5K
Because [user] likes [genre]	130.7K
Because it's popular	140.5K
Mood	140.7K
Focus	140.5K

$$r(j, e, x) = \sigma(\theta_{global} + \theta_j \times 1_j + \theta_e \times 1_e + \theta_x \times 1_x)$$

Why I like it: Multi-task Learning for Recommendation and Explanation [LDS18]

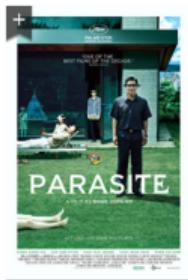
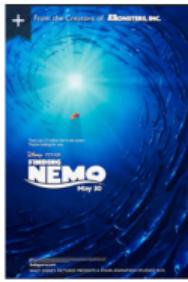


Если хотим делать объяснения рекомендаций, нужно ответить на вопросы:

- Какую цель мы достигнем объяснениями?
- Какие объяснения можно получить из модели?
- Как правильно представить объяснения пользователю?

Смещения

Удачные рекомендации



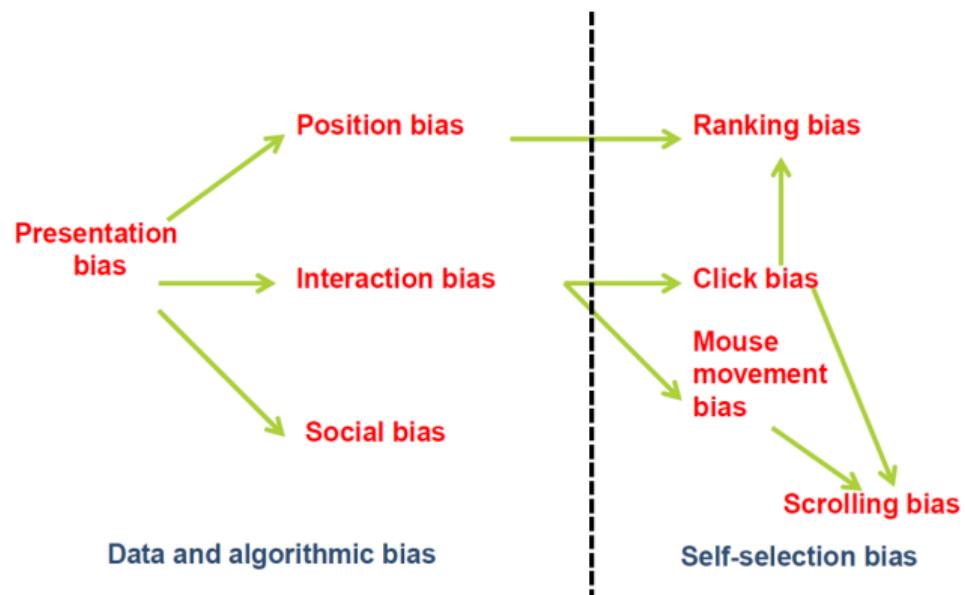
Пример self-selection bias

	Horror	Romance	Drama		Horror	Romance	Drama		Horror	Romance	Drama
Horror Lovers	5	1	3		p	p/10	p/2		1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
Romance Lovers	1	5	3		p/10	p	p/2		1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
	\hat{Y}_1	\hat{Y}_2	\hat{Y}_3		\hat{Y}_1	\hat{Y}_2	\hat{Y}_3		\hat{Y}_1	\hat{Y}_2	\hat{Y}_3
Horror Lovers	5	1	5		5	5	3		1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1
Romance Lovers	1	5	5		5	5	3		1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1

Figure 1. Movie-Lovers toy example. Top row: true rating matrix Y , propensity matrix P , observation indicator matrix O . Bottom row: two rating prediction matrices \hat{Y}_1 and \hat{Y}_2 , and intervention indicator matrix \hat{Y}_3 .

$$R(\hat{Y}) = \frac{1}{UI} \sum_u \sum_i \delta_{ui}(Y, \hat{Y}), \quad R_{naive}(\hat{Y}) = \frac{1}{N} \sum_{(u,i) \in D} \delta_{ui}(Y, \hat{Y})$$

Смещения в рекомендациях [Jar19]



Inverse Propensity Scored Estimator [SSS⁺16]

$P_{ui} = P((u, i) \in D)$ – вероятность, что пользователь u поставит оценку айтему i

$$R_{IPS}(\hat{Y}|P) = \frac{1}{UI} \sum_{(u,i) \in D} \frac{\delta_{ui}(Y, \hat{Y})}{P_{ui}}$$

$$\begin{aligned} E_D[R_{IPS}(\hat{Y}|P)] &= \frac{1}{UI} \sum_u \sum_i E_D \left[\frac{\delta_{ui}(Y, \hat{Y})}{P_{ui}} \mathbb{I}\{(u, i) \in D\} \right] = \\ &= \frac{1}{UI} \sum_u \sum_i \delta_{ui}(Y, \hat{Y}) = R(\hat{Y}) \end{aligned}$$

IPS Estimator: проблемы

1. Когда P_{ui} неизвестно, его приходится оценивать
2. Большая дисперсия при оценке P_{ui}
3. Непонятно, как быть с рекомендациями списков

Causal рекомендации

Традиционный рекомендер

Посмотрит ли пользователь этот фильм, если известно что она смотрела в прошлом?

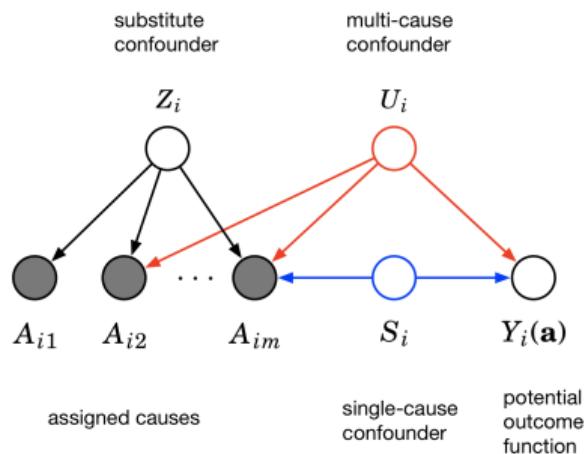
Causal рекомендер

Посмотрит ли пользователь этот фильм, если мы его порекомендуем, и известно, что она смотрела в прошлом?

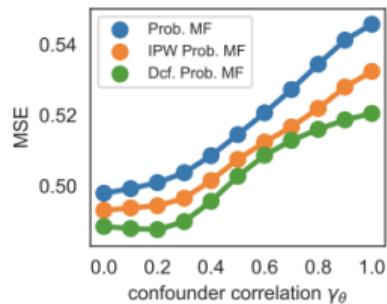
The Deconfounded Recommender [WLCB18]

Confounder

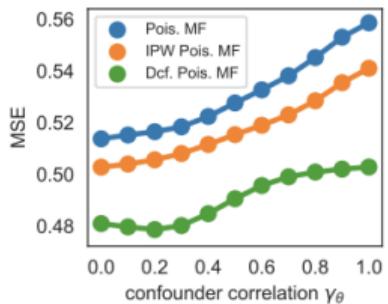
Переменная, которая влияет и на treatment assignment, и на outcome



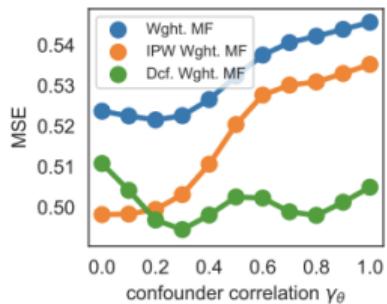
1. Учим модель $p(z, a_1, \dots, a_m)$
2. Оцениваем $E(z_j | a_{1j}, \dots, a_{mj})$ для каждого наблюдения
3. Используем оценки для z_j как признак в рекомендере



(a) Probabilistic MF



(b) Poisson MF



(c) Weighted MF

Из-за специфики сбора данных рекомендации подвержены смещениям.

Существуют техники для корректировки, но они несовершенны.

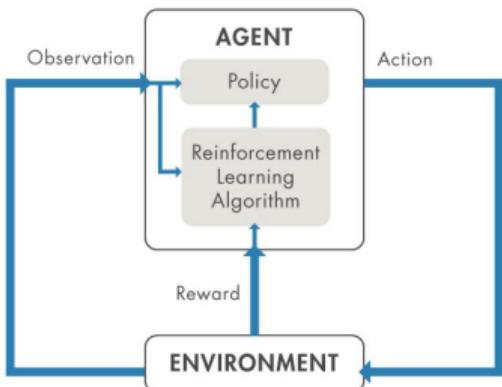
Долгосрочный эффект рекомендаций



Долгосрочный эффект рекомендаций

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

Рекомендации как Reinforcement Learning



RecSys	→	RL
Пользователь	→	Среда (environment)
Контекст	→	Наблюдение (observation)
Рекомендательный сервис	→	Агент (agent)
Алгоритм рекомендаций	→	Политика (policy)
Рекомендация	→	Действие (action)
Покупка, просмотр, клик	→	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

Разнообразие в рекомендательных системах
ooooooooo

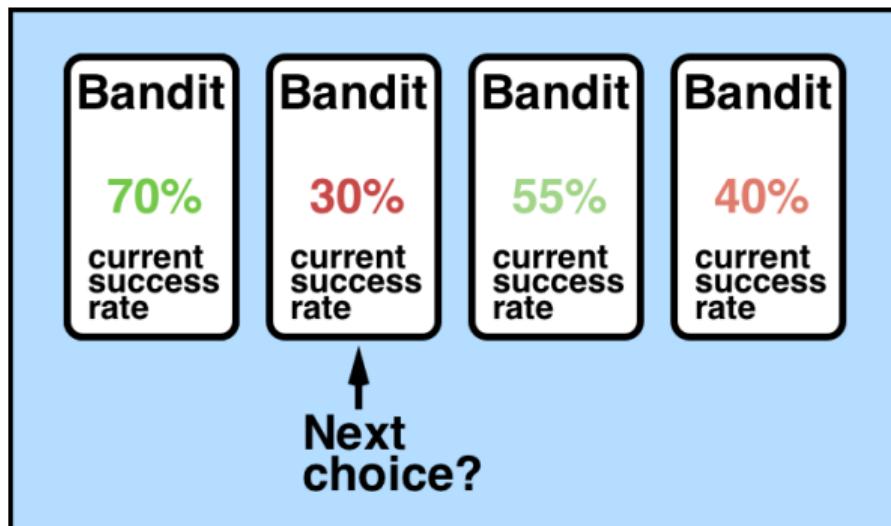
Объяснение рекомендаций
oooooooooooo

Смещения
oooooooooo ooooo

Долгосрочный эффект рекомендаций
●ooooooooo

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]

- ε -greedy: выбираем случайную руку с вероятностью ε , иначе жадно
- ε -decay: как ε -greedy, но уменьшаем ε 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

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

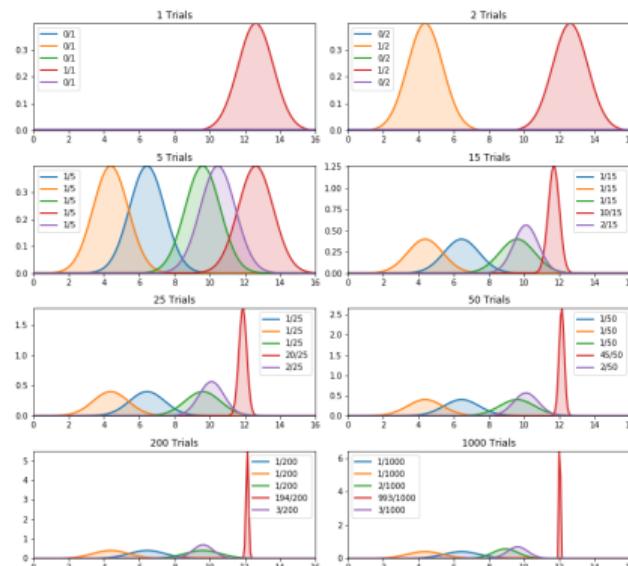
Обновление

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

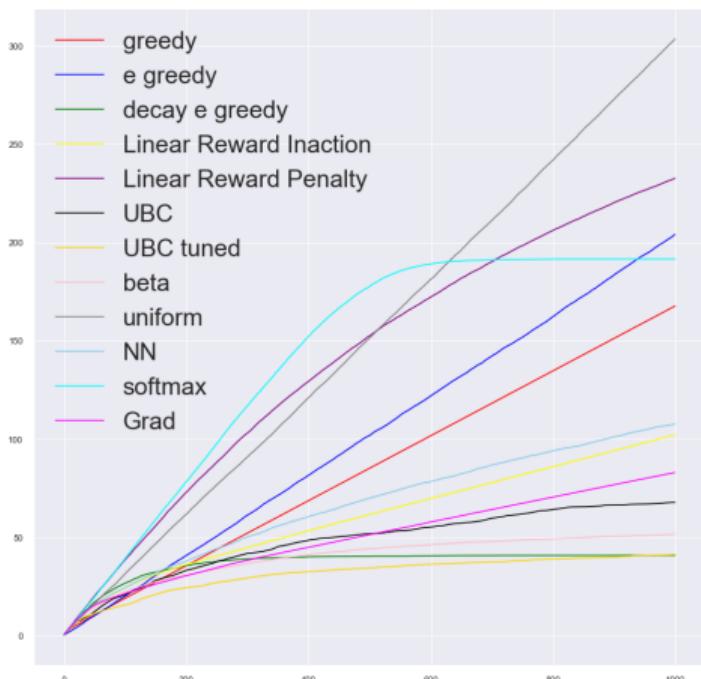
$$H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a), \quad \forall a \neq A_t$$

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

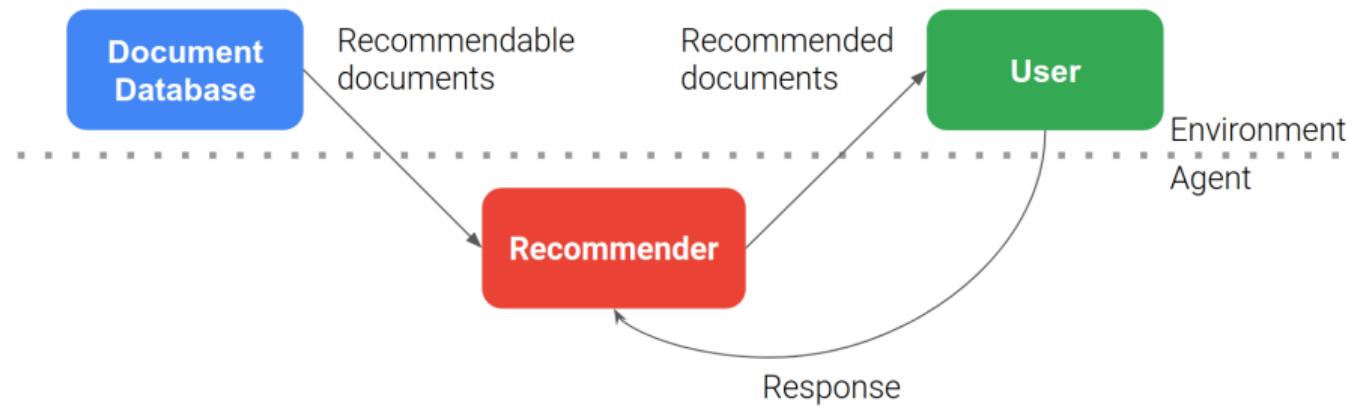


Сравнение алгоритмов [BAN19a]

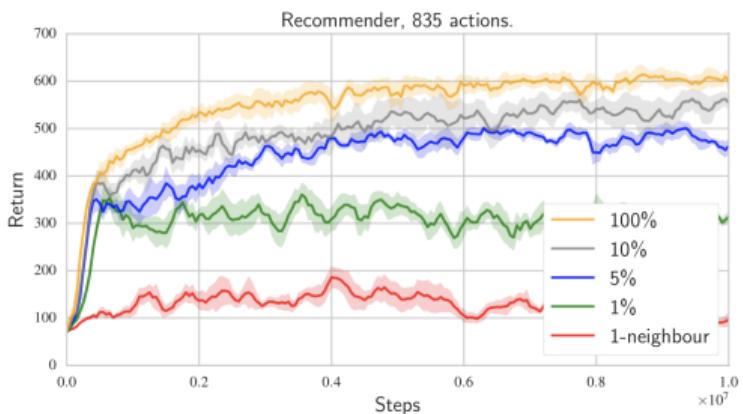
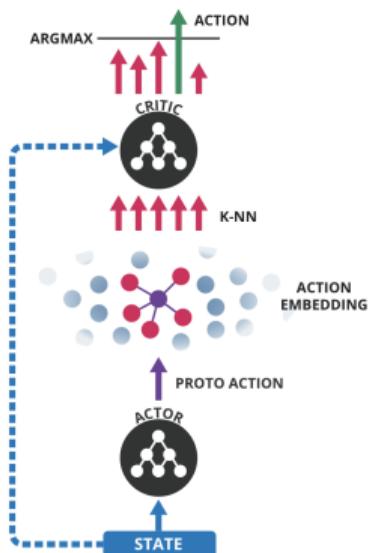


$$\text{Regret}_t = G_t^{\text{optimal}} - G_t$$

RecSim: A Configurable Simulation Platform for Recommender Systems [IHM⁺19]



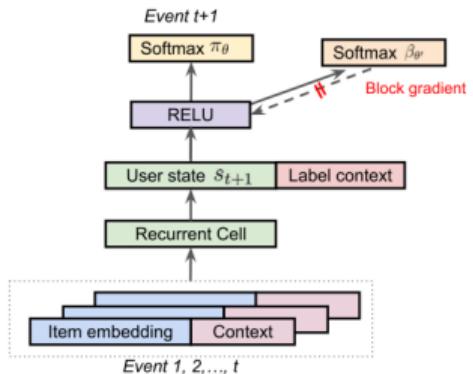
Deep Reinforcement Learning in Large Discrete Action Spaces [DAEH⁺15]¹



¹Пример использования в рекомендациях: <https://arxiv.org/abs/1811.05869>

Top-K Off-Policy Correction for a REINFORCE Recommender System [CBC⁺18]

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



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.

Разнообразие в рекомендательных системах
oooooooooo

Объяснение рекомендаций
oooooooooooo

Смещения
oooooooooo oooo

Долгосрочный эффект рекомендаций
oooooooooooo

Многорукие банды
oooooooooooo

Итоги

Итоги

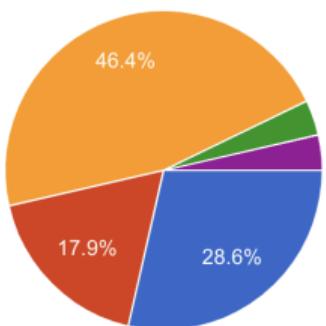
При построении моделей мы делаем упрощающие предположения. Из-за этих предположений в продакшен системах могут возникать негативные эффекты. Эти эффекты нужно учитывать и пытаться исправить.

Итоги курса

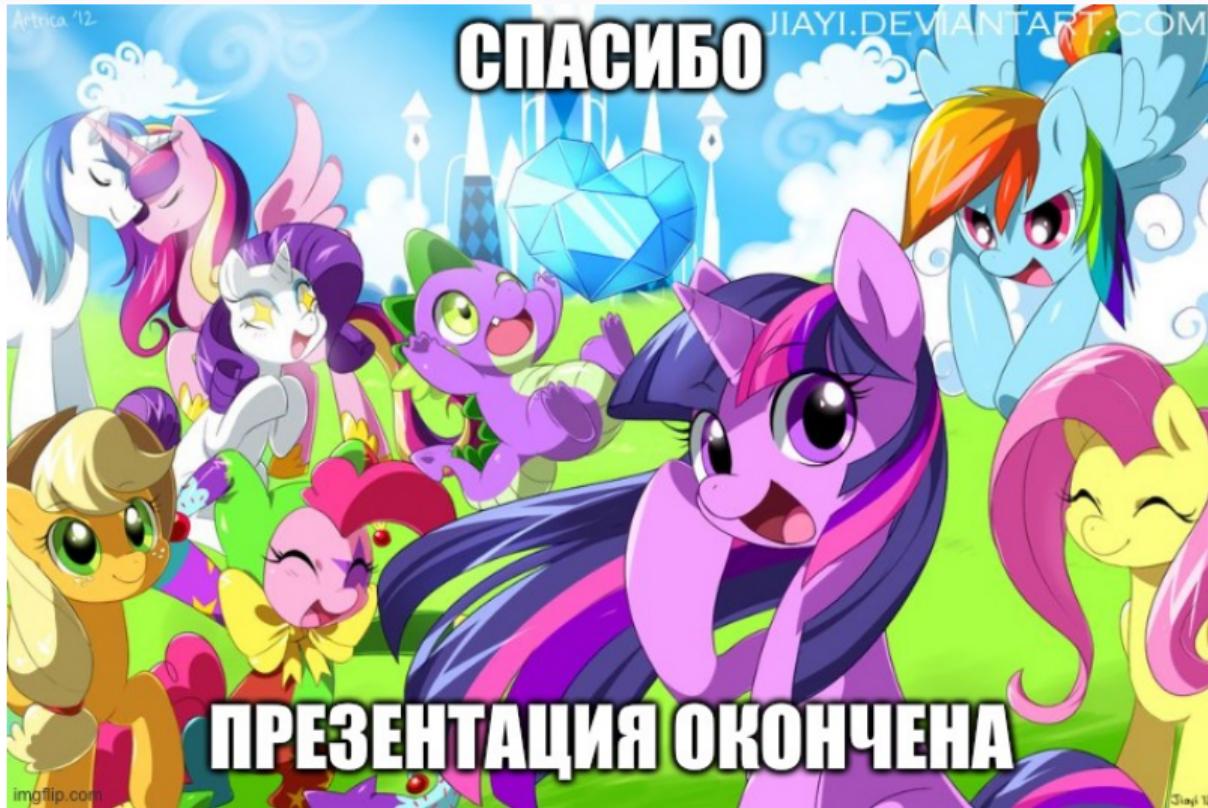
В будущем рекомендательные системы будут давать релевантные, разнообразные и полезные рекомендации. Они будут учитывать долгосрочные интересы пользователей. А пользователи будут понимать, почему им что-то предлагают и смогут контролировать механизмы построения рекомендаций.

Но понадобится ваша помощь. И научная честность.

Входной опрос



- Хочу получить базовое введение в рекомендательные сервисы
- Хочу получить глубокие теоретические знания о задаче реко...
- Хочу научиться создавать боевые рекомендательные сервисы
- Хочу просто пройти курс, это требуется, чтобы закончить МФТИ
- Первые три пункта: (базовое введение с интересным погружение...



Литература I

-  Mustafa Abdool, Malay Haldar, Prashant Ramanathan, Tyler Sax, Lanbo Zhang, Aamir Manaswala, Lynn Yang, Bradley Turnbull, Qing Zhang, and Thomas Legrand, *Managing diversity in airbnb search*, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '20, Association for Computing Machinery, 2020, p. 2952–2960.
-  *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](https://arxiv.org/abs/1812.02353) (2018).

Литература II

-  Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang, *Controllable multi-interest framework for recommendation*, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2020).
-  Gabriel Dulac-Arnold, Richard Evans, H. V. Hasselt, Peter Sunehag, Timothy P. Lillicrap, Jonathan J. Hunt, Timothy A. Mann, Théophane Weber, Thomas Degrif, and Ben Coppin, *Deep reinforcement learning in large discrete action spaces*, arXiv: Artificial Intelligence (2015).
-  Eugene le, 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.
-  Greg Jarboe, *Understanding biases in search and recommender systems*, 2019.

Литература III

-  Mesut Kaya and Derek Bridge, *A comparison of calibrated and intent-aware recommendations*, Proceedings of the 13th ACM Conference on Recommender Systems (New York, NY, USA), RecSys '19, Association for Computing Machinery, 2019, p. 151–159.
-  Yichao Lu, Ruihai Dong, and Barry Smyth, *Why i like it: Multi-task learning for recommendation and explanation*, Proceedings of the 12th ACM Conference on Recommender Systems (New York, NY, USA), RecSys '18, Association for Computing Machinery, 2018, p. 4–12.
-  James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra, *Explore, exploit, and explain: Personalizing explainable recommendations with bandits*, Proceedings of the 12th ACM Conference on Recommender Systems (New York, NY, USA), RecSys '18, Association for Computing Machinery, 2018, p. 31–39.

Литература IV

-  Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims, *Recommendations as treatments: Debiasing learning and evaluation*, Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, p. 1670–1679.
-  Yixin Wang, Dawen Liang, Laurent Charlin, and David M. Blei, *The deconfounded recommender: A causal inference approach to recommendation*, ArXiv [abs/1808.06581](https://arxiv.org/abs/1808.06581) (2018).