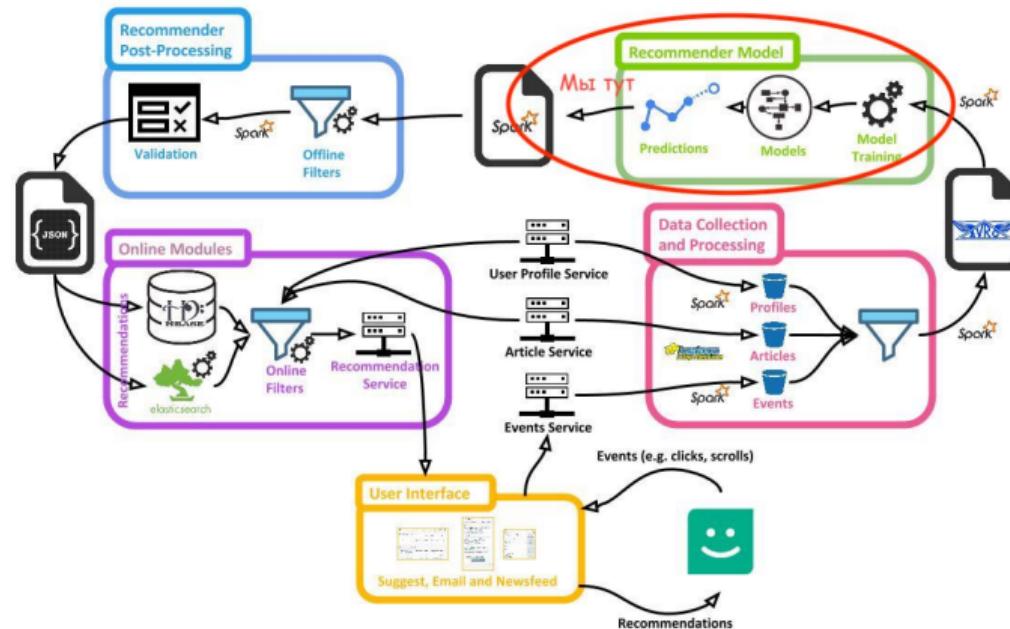


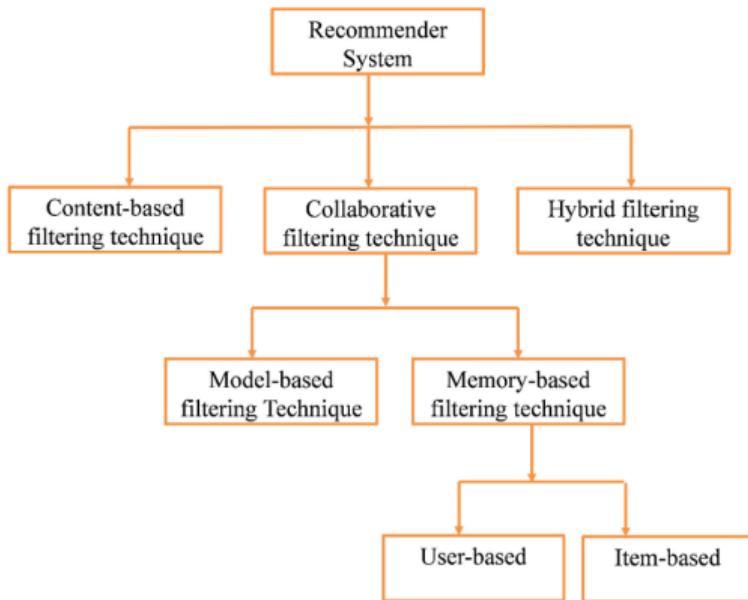
Классические алгоритмы рекомендаций 1

Петр Погорелов

5 марта 2025 г.

Контекст





Content-based RS
●oooooooooo

Neighbourhood-based Collaborative Filtering
oooooooooooo

Content-based RS



Пример: интуиция

Нравится:

- Возвращение короля
 - Король былого и грядущего
 - Война мага

Не нравится:

- Новый ум короля

Что порекомендуем?

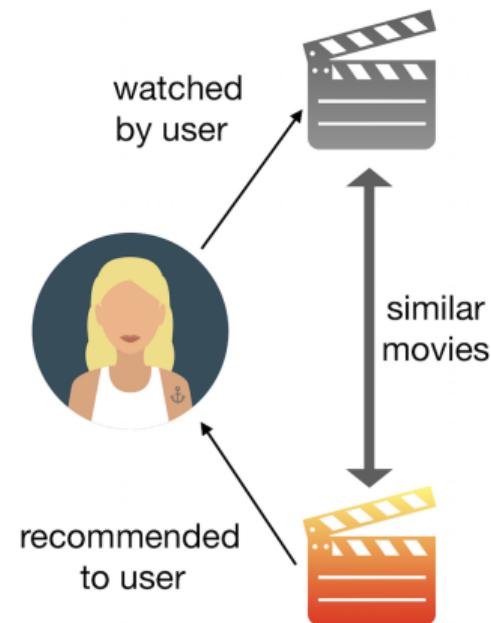
- Битва королей
 - Война и мир



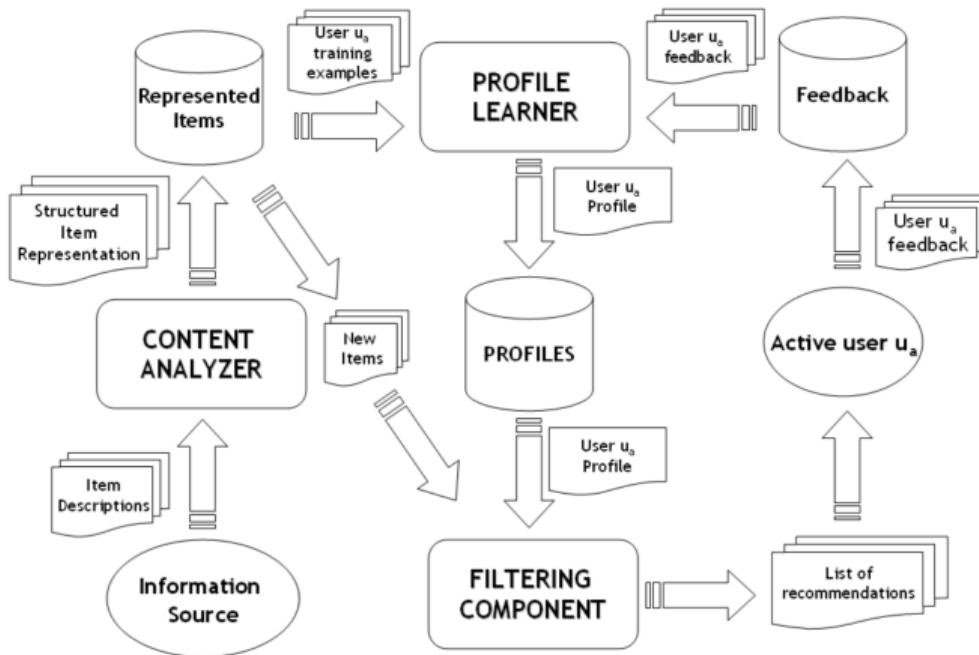
Content-based RS

Идея

Рекомендуем пользователю айтемы, похожие на те, что нравились ей раньше



Архитектура CBRS [?]



Анализ контента

Данные	Признаки
Табличные	Категориальные / числовые
Текст	BOW / TF-IDF / BM25 / NN Эмбеддинги
Картинки	SIFT / SURF / NN Эмбеддинги
Музыка	Spectral



Supervised профили пользователей

Модель

$$p(u \text{ likes } i) = f(x_i, x_u, \theta)$$

$$\text{recommendations} = \arg; \text{topk } p(u \text{ likes } i)$$

Обучаемые параметры:

- x_u – профиль пользователя
- θ – параметры модели

Данные:

- $\{(x_i, u_j \text{ likes } x_i)\}^N$

Примеры моделей:

- Naive Bayes
- Любой классификатор



Unsupervised профили пользователей

Идея

Храним айтемы, с которыми взаимодействовал пользователь, и рекомендуем ближайшие к ним.

Когда айтемов у пользователя слишком много:

- Храним последние
- Кластеризуем и храним представления кластеров [?]



Пример: формально

Naive Bayes

Нравится:

- Возвращение короля
- Король былого и грядущего
- Война мага

Не нравится:

- Новый ум короля

Что порекомендуем?

1. Битва королей
2. Война и мир

$$\begin{aligned} p(c|d) &\sim p(c)p(d|c) = \\ &= p(c) \prod_j p(w_j|c) \sim \log p(c) + \sum_j \log p(w_j|c) \\ p(w_j|c) &= \frac{N_{jc} + \alpha}{N_c + \alpha|V|} \end{aligned}$$

Размер словаря $|V| = 10$, $\alpha = 1$

Вероятности классов $p(+) = 3/4$, $p(-) = 1/4$

Скоры документов

$$p(+|1) \sim \log 3/4 + \log 1/13 + \log 3/13$$

$$p(-|1) \sim \log 1/4 + \log 1/11 + \log 2/11$$

$$s(1) = p^*(+|1) - p^*(-|1) = 2.69$$



Известные использования

- Spotify: Deep content-based music recommendation [?]
- Ozon: Векторное представление товаров Prod2Vec: как мы улучшили матчинг и избавились от кучи эмбеддингов [?]

Итоги

Плюсы

- Рекомендации строятся независимо для каждого пользователя
- Рекомендации часто можно объяснить
- Естественная поддержка холодного старта айтемов

Минусы

- Полагаются на (несовершенные) техники анализа контента
- Нет поддержки холодного старта пользователей
- Отсутствие новизны: умеют рекомендовать только похожие айтемы



Content-based RS
oooooooooo

Neighbourhood-based Collaborative Filtering
●oooooooooooo

Neighbourhood-based Collaborative Filtering



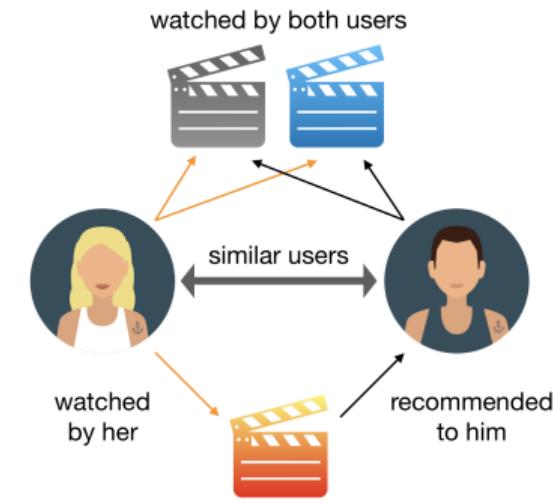
Пример: интуиция [?]

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4	?	?	2	
C				2	4	5	
D		3					3

Collaborative filtering¹-based RS

Идея

Рекомендуем пользователю айтемы, которые понравились похожим на нее пользователям. Пользователи похожи, если они похоже оценивают одни и те же айтемы.



¹Оскар за худшее название алгоритма

User-based

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{v \in N_i(u)} w_{uv} h(r_{vi})}{\sum_{v \in N_i(u)} w_{uv}} \right)$$

Item-based

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{j \in N_u(i)} w_{ij} h(r_{uj})}{\sum_{j \in N_u(i)} w_{ij}} \right)$$

- $N_i(u)$ – соседи пользователя u , которые оценили айтем i
- $N_u(i)$ – соседи айтема i , которые оценила пользователь u
- w_{uv}, w_{ij} – веса соседей
- h – функция нормализации

Как вычислить веса w_{uv} , w_{ij} ?

$$\cos(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2 \sum_{i \in I_v} r_{vi}^2}}$$

$$\text{pearson}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi}^2 - \bar{r}_v^2)}}$$

Дано:

100 айтемов

1000 пользователей

10000 рейтингов равномерно распределены по пользователям и айтемам

Вопрос:

Сколько в среднем общих айтемов у пары пользователей?

Сколько в среднем общих пользователей у пары айтемов?

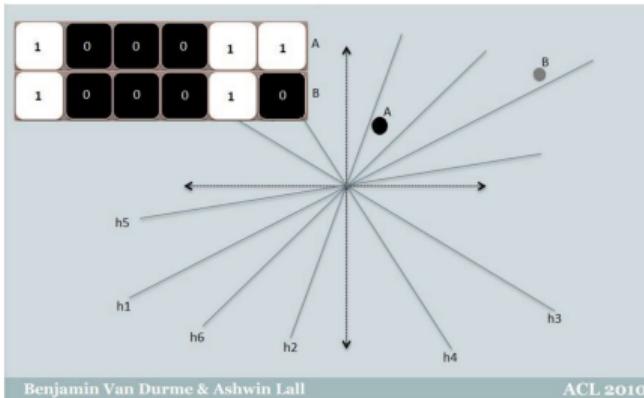
Небольшое количество надежных соседей лучше, чем много ненадежных

- User-based ($|U| < |I|$)
- Item-based ($|U| > |I|$)



Locality-Sensitive Hashing для приближенного поиска соседей

The general idea of LSH is to use a family of functions ("LSH families") to hash data points into buckets, so that the data points which are close to each other are in the same buckets with high probability, while data points that are far away from each other are very likely in different buckets.



Ha Spark

Bucketed Random Projection for Euclidean Distance

<https://spark.apache.org/docs/2.2.3/ml-features.html#bucketed-random-projection-for-euclidean-distance>

MinHash for Jaccard Distance

<https://spark.apache.org/docs/2.2.3/ml-features.html#minhash-for-jaccard-distance>



Пример: формально

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4	?	?	2	
C				2	4	5	
D		3					3

Нормализация: $h(r) = r$ Веса

$$\cos(A, B) = \frac{45}{\sqrt{42}\sqrt{66}} = 0.37$$

$$\cos(B, C) = ?$$

$$r(TW) = ? \quad r(SW1) = ?$$



Плюсы

- Простота и интуитивность: рекомендации можно объяснить.
- Небольшое количество параметров
- Не нужно обучать, удобно добавлять новых пользователей и айтемы

Минусы

- User-based: очень много пользователей для поиска NN
- Item-based: как понять, для каких айтемов считать рейтинги?
- Разреженность пространства





Литература I

-  Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman, *Mining of massive datasets*, 2nd ed., Cambridge University Press, USA, 2014.
-  *Векторное представление товаров prod2vec: как мы улучшили матчинг и избавились от кучи эмбеддингов.*
-  Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec, *Pinnersage: Multi-modal user embedding framework for recommendations at pinterest*, 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. 2311–2320.
-  Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, *Recommender systems handbook*, 1st ed., Springer-Verlag, Berlin, Heidelberg, 2010.

Литература II

-  Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen, *Deep content-based music recommendation*, Advances in Neural Information Processing Systems (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.