

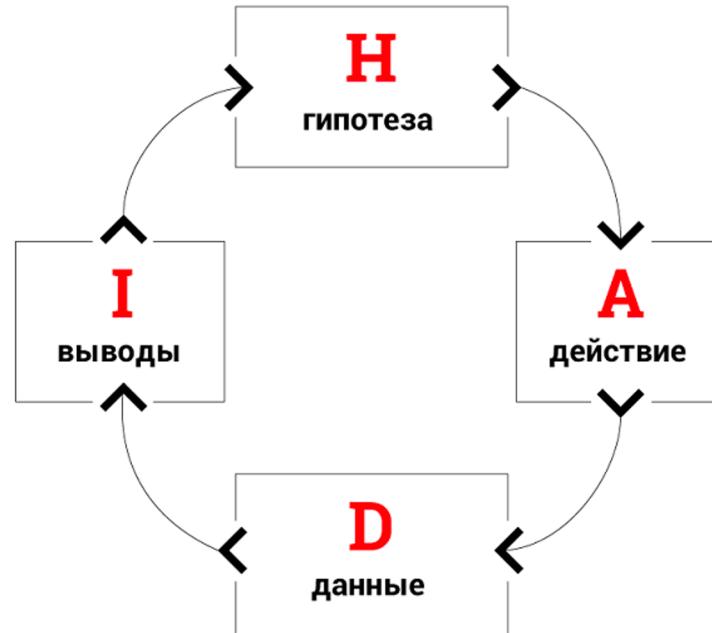
AI Talent Hub Baselines and kNN

Елисова Ирина
ML Engineer
MTC BigData

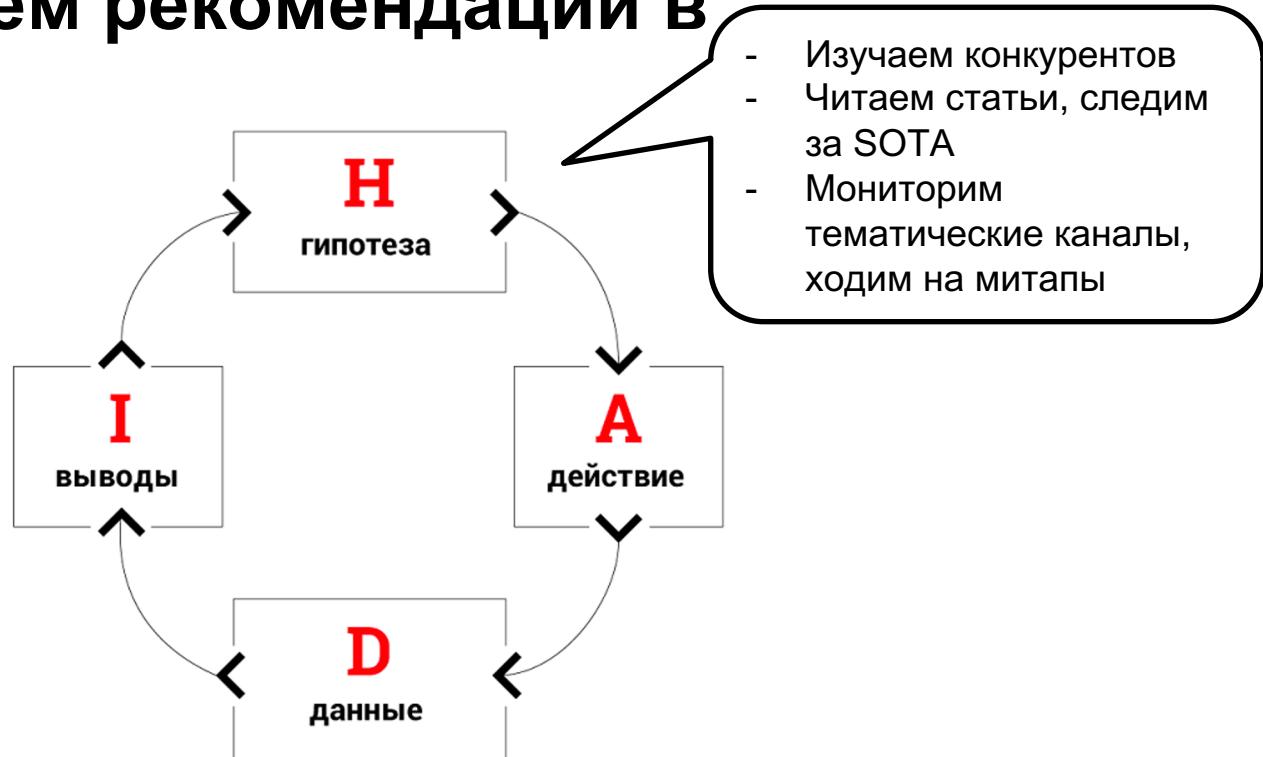


- 1. Процесс развития
рекомендаций в сервисе**
- 2. Бейзлайны и эвристики**
- 3. kNN модели**
- 4. Другие базовые модели**

Как улучшаем рекомендации в сервисе?



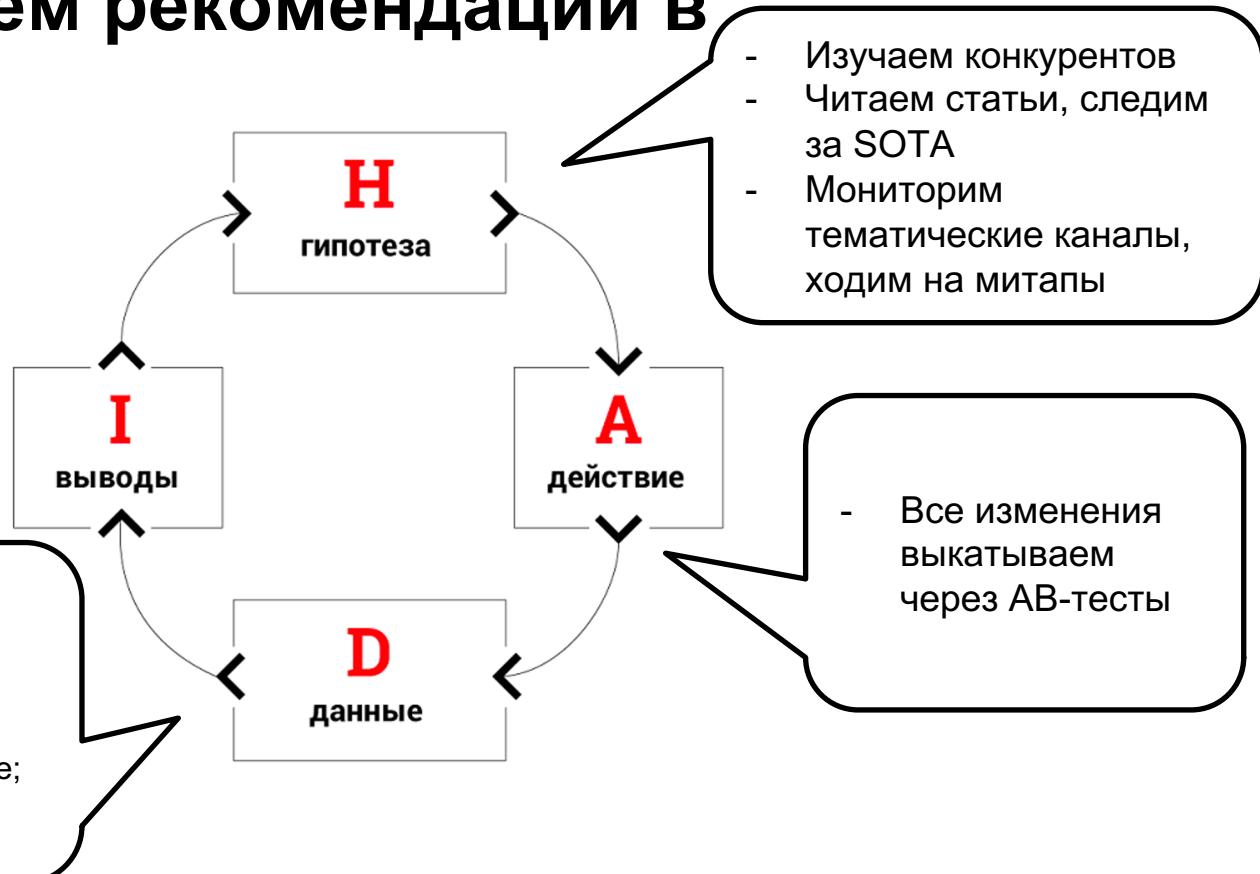
Как улучшаем рекомендации в сервисе?



Как улучшаем рекомендации в сервисе?



Как улучшаем рекомендации в сервисе?



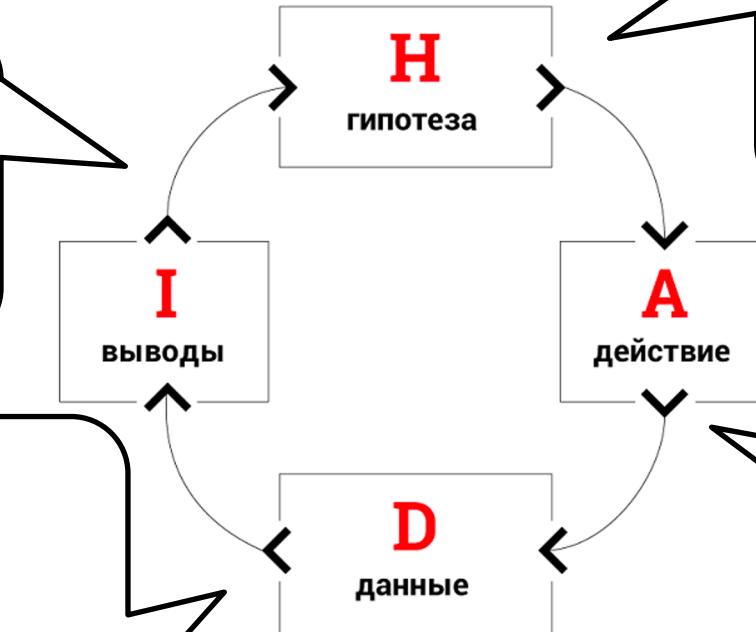
Как улучшаем рекомендации в сервисе?

- Всегда знаем что дальше улучшаем: пул гипотез сформирован

- Изучаем конкурентов
- Читаем статьи, следим за SOTA
- Мониторим тематические каналы, ходим на митапы

- Корректно готовим и оцениваем АБ-тест

одно изменение за раз;
незаинтересованность
анализирующего в результате;
коррекция на множественную
проверка гипотез;



- Все изменения выкатываем через АВ-тесты

Бейзлайны

Вам выгрузили просмотры
пользователей за месяц и
сказали придумать
рекомендательную систему
до завтра.

Ваши действия?



Начинаем с топа популярных айтемов

Вам выгрузили просмотры пользователей за месяц и сказали придумать рекомендательную систему до завтра.

Ваши действия?



.groupby(item)
.count()
.top(N)

Персонализируем популярное

Пять ваших менеджеров
посмотрели свои
рекомендации и заметили,
что они одинаковые.

А как же персонализация?

Персонализируем популярное

Пять ваших менеджеров
посмотрели свои
рекомендации и заметили,
что они одинаковые.

А как же персонализация?

Разбиваем топ популярного
по категориям на несколько
вариантов

Персонализируем популярное

Пять ваших менеджеров
посмотрели свои
рекомендации и заметили,
что они одинаковые.

А как же персонализация?

**Разбиваем топ популярного
по категориям на несколько
вариантов**

Например соцдем:
пол, возраст



Вариации топа популярных айтемов

- Разные варианты агрегации данных (count, mean, median, ...)
- Период
- Сэмплирование

[rectools.models.popular.PopularModel](#)

[rectools.models.popular_in_category.PopularInCategoryModel](#)

Вариации топа популярных айтемов по покрытию уникальных пользователей

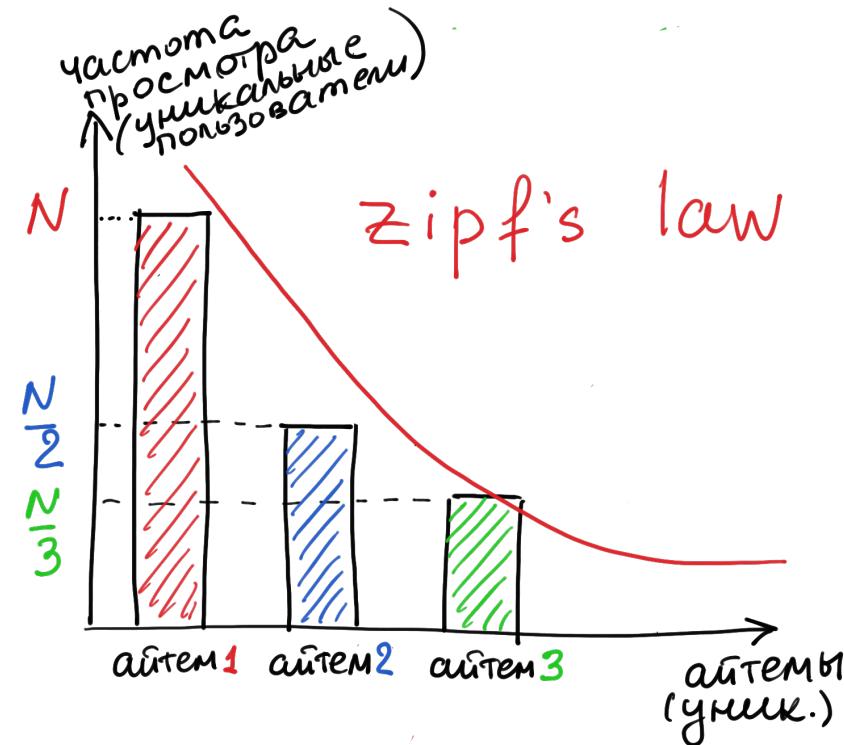
Ищем топ айтемов, которые
бы посмотрело n%
уникальных пользователей

Вариации топа популярных айтемов

по покрытию уникальных пользователей

Ищем топ айтемов, которые бы посмотрело $n\%$ уникальных пользователей

1. Выбираем самый популярный айтем по количеству пользователей
2. Исключаем этих пользователей
3. Ищем следующий самый популярный айтем
4. Продолжаем пока не переберем всех пользователей (допущение – zipf's law)



Достоинства топа популярного

- Хороший бейзлайн
- Быстро считается
- Быстро выкатить в продакшн
- Решение проблемы cold start
- Решение технических проблем невыдачи рекомендаций

Недостатки топа популярного

- Низкий coverage, хвост распределения никогда не будет показан
- Релевантный, но непопулярный айтем никогда не будет показан пользователю
- Не попадают новинки, еще не набравшие просмотров
- “Ловушка популярного как бейзлайна”: bias в оффлайн валидации

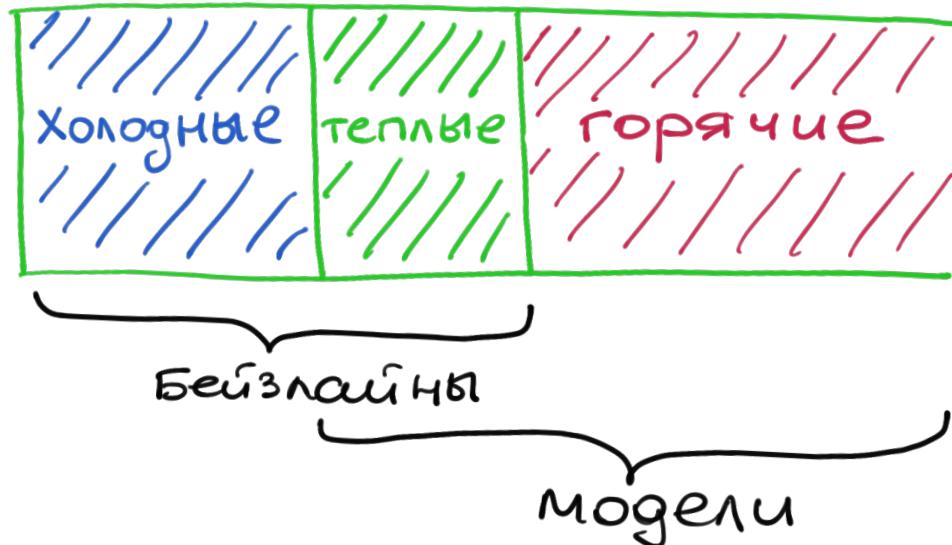
Техническая сторона бейзлайна ака “заглушка”

Должен всегда считаться, независимо от
других моделей и выдаватьсь пользователю
когда *что-то не сработало*

- модель выдала рекомендаций меньше N
- модель не обновилась
- фильтры порезали выдачу рекомендаций
- your fail option here

[ItemItemRecommender can generate less than N recommendations issue](#)

Бейзлайн: для каких пользователей



Эвристики следующий шаг

Эвристики и бизнес-правила

В каждом домене - свои бейзлайны, основанные на эвристиках

Примеры

- Топ Кинопоиска или IMDB для онлайн-кинотеатра
- Последние купленные товары (в наличии) в ритейле
- Матрица переходов для продажи электроники (цепь маркова 1го порядка). С какого на какой девайс переходит человек: устаревшие модели не рекомендуем

Важно отличать правила-бейзлины от правил пост- и пре-процессинга рекомендаций

В реальном проекте всегда будет набор бизнес-правил, относящихся к процессингу рекомендаций

Примеры

- К дню святого валентина добавить в подборку фильмы про любовь
- Как рекомендовать сериалы? Серию или сезон?
- Добавление рекламных постов в ленту

Важно отличать правила-бейзлайны от правил пост- и пре-процессинга рекомендаций

Важно: все правила необходимо мониторить, вовремя удаляя устаревшие и создавая новые

Минусы

- поддержка актуальности
- немасштабируемость

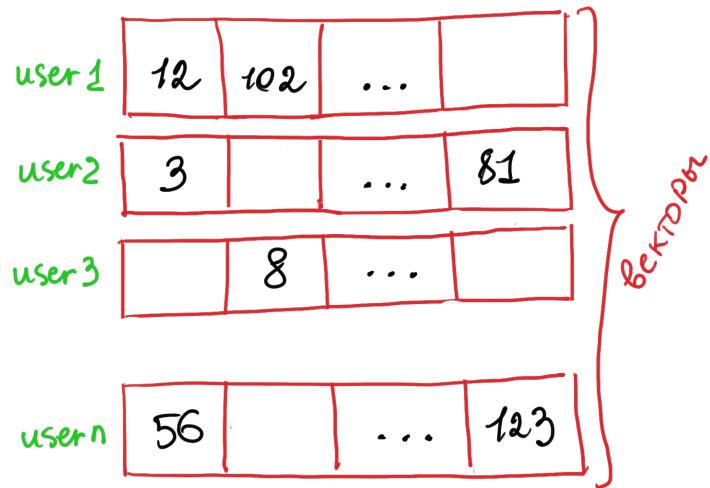
Модели на ближайших соседях userkNN , itemkNN

Основная концепция kNN: расстояния между векторами

| | item1 | item2 | ... | item m |
|--------|-------|-------|-----|--------|
| user1 | 12 | 102 | ... | ? |
| user2 | 3 | ? | ... | 81 |
| user3 | ? | 8 | ... | ? |
| ... | ... | ... | ... | ... |
| user n | 56 | ? | ... | 123 |

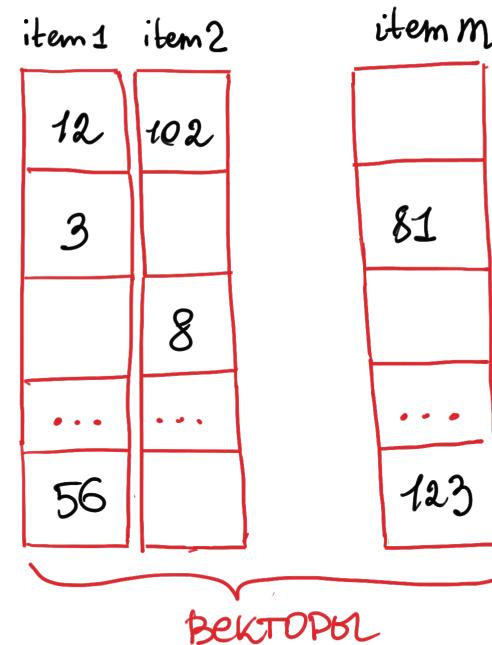
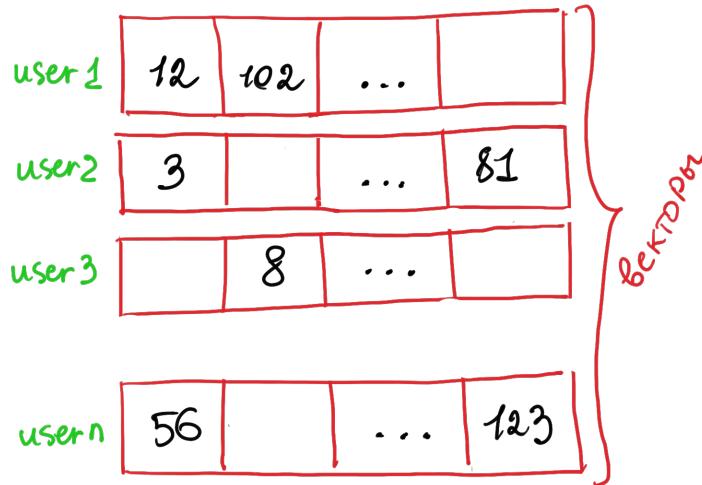
База взаимодействий user – item → матрица user – item →

Основная концепция kNN: расстояния между векторами



База взаимодействий user – item → матрица user – item → вектор

Основная концепция kNN: расстояния между векторами

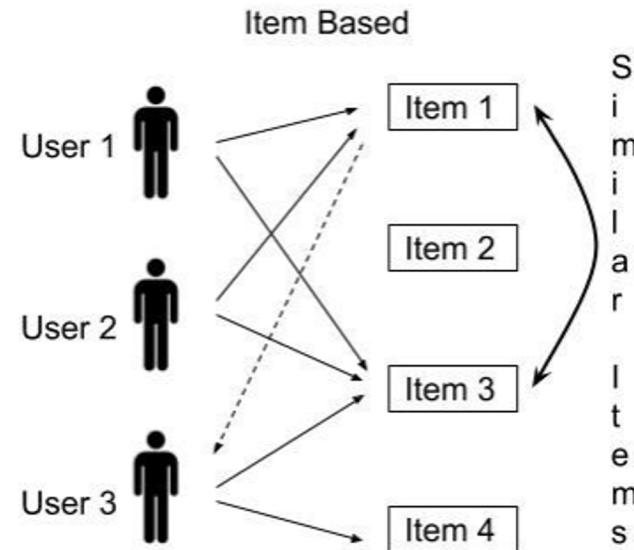


База взаимодействий user – item → матрица user – item → вектор →
его ближайшие соседи → рекомендации

itemkNN

item-based = ищем похожие айтемы

- Считаем близость всех айтемов со всеми
- Предлагаем пользователю айтемы, которые похожи на те, с которыми он уже повзаимодействовал



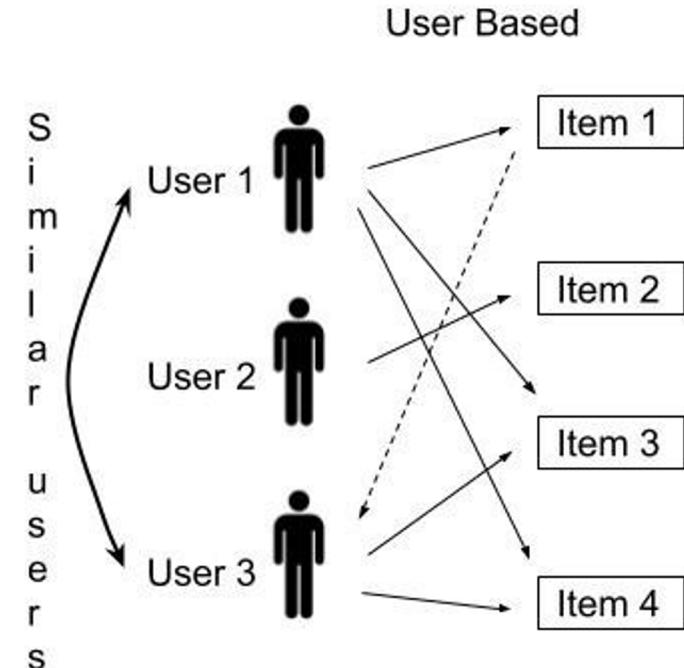
userkNN

user-based = ищем похожих пользователей

- Считаем близость всех пользователей со всеми

[Нормируем на среднюю оценку по пользователю, так как у всех разное видение на постановку оценок]

- Ищем ближайших пользователей к этому пользователю
- У них смотрим айтемы, которые этот пользователь еще не смотрел



Плюсы подхода

- Хорошо работает, хороший **бейзлайн**
- **Скорость** обучения
- **Скорость** инференса
- Возможность **интерпретации** результатов
«Похожим на вас пользователем нравится...»
- Возможность закрыть одной моделью **несколько типов рекомендаций** (+ item2item рекомендации)

Минусы подхода

- **Матрицу близости надо пересчитывать**
 - появляются новые айтемы / пользователи
- **Проблема холодного старта**
- **Нет учета временной** составляющей
- **Если взаимодействий мало**, то векторы могут оказаться ошибочно близкими (пример: пользователи посмотрели только пару популярных общих фильмов и модель стала считать их похожими)

Как можно измерять близость векторов

- косинусное расстояние
- корреляция Пирсона
- расстояние Хэмминга
- Манхэттенское расстояние

[70+ формул близости для бинарных векторов](#)

Можно взвешивать исходные матрицы взаимодействий

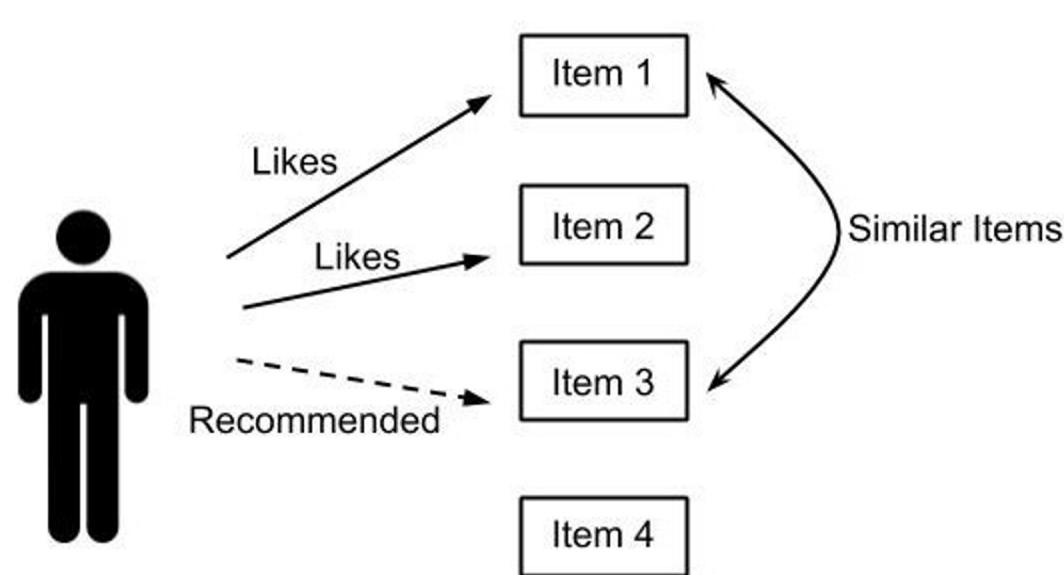
- TF-IDF
- BM25

kNN: что дальше?

- **item2item** рекомендации
- **item2item -> user2item**
 - Джоиним item2item к просмотрам
 - Ранжируем их с весами (можно учесть временную составляющую)
- **Как генерация кандидатов для модели второго уровня**

Другие базовые модели

Content-based подход



Sparse Linear Method (SLIM)

$$\tilde{A} = A \cdot W$$

Учим sparse матрицу коэффициентов W ,
Получаем новую матрицу коэффициентов

A user-item матрица бинарных взаимодействий

W sparse матрица коэффициентов W

\tilde{A} матрица с рейтингами, которую предсказываем. Ее конкретная строка - скоры айтемов для рекомендаций этого пользователя. Сортируем (новые для пользователя) айтемы по скору → получаем рекомендации

$$\tilde{a}_{ij} = \vec{a}_i^T \cdot \vec{w}_j$$

$$\vec{w}_j$$

предсказанный скор

вектор-столбец коэффициентов

Sparse Linear Method (SLIM)

$$\tilde{A} = A \cdot W$$

Метод хорошо работает, когда мало истории взаимодействий

Учим sparse матрицу коэффициентов W
через решение оптимизационной задачи

$$\frac{1}{2} \left\| A - \underbrace{\tilde{A}W}_{\tilde{A}} \right\|_F^2 + \underbrace{\frac{\beta}{2} \|W\|_F^2 + \lambda \|W\|_1}_{\text{регуляризация}} \rightarrow \min_W$$

$\|\cdot\|_F$ Норма Фробениуса (вспоминаем операции над матрицами)

$\|\cdot\|^2$ L2 норма

$\|\cdot\|_1$ L1 норма

Frequent Pattern Mining

Association rule learning

$\{\text{масло, хлеб}\} \Rightarrow \{\text{молоко}\}$

$X \Rightarrow Y$
ассоциативное
правило

Впервые появилось
в базах данных
ритейла

$D = \{d_1, \dots, d_n\}$ множество транзакций пользователя

$I = \{i_1, \dots, i_m\}$ множество айтемов в транзакциях

$X, Y \subseteq I$ наборы айтемов, из которых строится
правило if ... then

Frequent Pattern Mining

Association rule learning

support = как часто айтем появляется в данных

$$\text{Support} = P(X \cap Y) = \frac{\text{кол-во транзакций}}{\text{всего транзакций}} \quad \begin{matrix} \text{кол-во транзакций} \\ \text{в к-х есть } X \text{ и } Y \end{matrix}$$

обычно задается *minimum support*

confidence = сколько раз правило if-then было верным

$$\text{confidence} = P(Y|X) = \frac{\text{support}(X \cap Y)}{\text{Support}(X)} = \frac{\text{кол-во транзакций}}{\text{кол-во транзакций с } X} \quad \begin{matrix} \text{кол-во транзакций} \\ \text{в к-х есть } X \text{ и } Y \end{matrix}$$

Frequent Pattern Mining в рекомендациях

кейс Intake24

Сервис логирования употребляемой еды – рекомендации в подсказках (для быстроты и удобства ввода)

База 5к айтемов

3 алгоритма

- Association rules
- Transactional item confidence
- Pairwise association rules

Цепи Маркова: случайные процессы

$$P(\text{будущее состояние} \mid \text{текущее состояние, прошлое состояние}) =$$

$$= P(\text{будущее состояние} \mid \text{текущее прошлое состояние, сост.})$$

Марковское
свойство

распределение вероятностей **следующего состояния**
(что юзер сделает дальше) зависит от **текущего**
состояния, но **не от прошлых** состояний

Цепи Маркова в рекомендациях

Кейс: матрица переходов для девайсов

$$D = \{D_1, D_2, D_3\}$$

переход
на
девайс 1 девайс 2 девайс 3

Вер-ть
перейти
с девайса 1

$$P = \begin{pmatrix} & \text{на} & \text{на} & \text{на} \\ & \text{нечто} & \text{дев. 2} & \text{дев. 3} \\ \text{на} & 0,1 & 0,8 & 0,1 \\ \text{нечто} & 0,75 & 0,05 & 0,2 \\ \text{дев. 1} & 0,5 & 0,3 & 0,2 \end{pmatrix}$$

матрица
переходных
вероятностей

Что купит
пользователь в
следующий раз?

Цепи Маркова в рекомендациях

Кейс: матрица переходов для девайсов

Исходная вероятность покупки одного из трех девайсов

$$q_0 = (0,05; 0,8; 0,15)$$

Вычисляем для каждого пользователя вероятность перехода на конкретный девайс в следующий раз:

$$q_1 = q_0 \cdot P = (0,68; 0,125; 0,195)$$

Что купит
пользователь в
следующий раз?

Итого

- Обсудили как развивать рекомендательную систему
- Разобрали бейзлайн популярного и эвристики как хороший старт и как заглушку
- kNN модели и другие базовые модели: SLIM, FP-mining, Марковские цепи как следующий шаг

Реализации

RecTools (Popular, kNN)

Implicit (kNN)

RecBole (Popular, kNN)

mlextend (FP-mining)

Mllib (spark) (FP-mining)

SLIM

Полезные ссылки

- [Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches](#)
 - [Netflix TechBlog, Spotify Research](#)
-
- [Item-based top- N recommendation algorithms](#)
 - [Distance Metrics from Ben Fredrickson \[implicit author\]](#)
 - [A Survey of Binary Similarity and Distance Measures](#)
 - [Locality-Sensitive Hashing for Finding Nearest Neighbors](#)
-
- [SLIM: Sparse Linear Methods for Top-N Recommender Systems](#)
 - [Han et al., Mining frequent patterns without candidate generation](#)
 - [Association Rules and Offline-Data-Based Recommender Systems](#)
 - [Краткое введение в цепи Маркова](#)