



Рекомендательные системы

Лекция №9

Журавлёв Вадим

Показывать

Ближайшие две недели Весь семестр

Дисциплина

Основы машинного обучения

Тип события

Все типы

Группа

Все группы

30 сентября 18:00 – 21:00 среда Основы машинного обучения Смешанное занятие 1 Уточняется ML-11

7 октября 18:00 – 21:00 среда Основы машинного обучения Смешанное занятие 2 Уточняется ML-11

14 октября 18:00 – 21:00 среда Основы машинного обучения Смешанное занятие 3 Уточняется ML-11

21 октября 18:00 – 21:00 среда Основы машинного обучения Смешанное занятие 4 Уточняется ML-11

28 октября 18:00 – 21:00 среда Основы машинного обучения Смешанное занятие 5 Уточняется ML-11

3 ноября вторник 18:00 – 21:00 Основы машинного обучения Смешанное занятие 6 Уточняется ML-11

11 ноября среда 18:00 – 21:00 Основы машинного обучения Смешанное занятие 7 Уточняется ML-11

сентябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

октябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

ноябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс

Приходя на лекцию



отметиться не
забудь ты

Содержание занятия

1. Обзор задач для рекомендательных систем
2. Данные
3. Виды рекомендательных систем
4. Семинарская часть
5. Метрики

Постановка задачи

Требуется порекомендовать пользователю что-то (товар, услугу, продукт, контент), что сможет его заинтересовать.



Рекомендательные системы

**Зачем и кому
нужны?**

Рекомендательные системы для организаций

- max вероятность покупки
- max матожидание прибыли
- товары из категории (long-tail)

Рекомендательные системы для пользователей

Актуальность рекомендаций обусловлена разнообразием товаров и услуг таким, что пользователь не в состоянии ознакомиться со всем ассортиментом за адекватное время

Предмет рекомендации

Что можно рекомендовать?

Предмет рекомендации

Товары	Книги Фильмы Музыка Игры Приложения
Контент	Новости Сайты Статьи Видео
Досуг	Рестораны Отели Театральные представления Выставки Туры
Социальные связи	друзья группы

Предмет рекомендации

Что рекомендовать?

- Товары (Amazon, Ozon, Aliexpress)
- Статьи (Arxhiv.org)
- Новости (Surfingbird, Яндекс.Дзен, Пульс)
- Изображения
- Видео (YouTube, Netflix)
- Люди (Linkedin, VK)
- Музыка (Yandex.Music, Spotify)

В целом рекомендовать можно что угодно

Netflix Prize

Задача:

- Необходимо улучшить алгоритм рекомендации фильмов на Netflix на 10%

Обучающие данные:

- 100 490 507 оценок (от 1 до 5)
- 480 189 пользователей
- 17 770 фильмов (Названия и год выхода в прокат)
- Дата оценок

Предсказать:

- 2 800 000 оценок (float)

Метрика:

- RMSE



Netflix Prize

На открытой выборке:

№	Команда	СКО	Улучшение в %	Время отправки
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28

На скрытой выборке обе команды улучшили результаты на **10.06 %**

Какие входные данные?

#013

Какие входные данные?

- Users (Пользователи)
- Items (Товары, Музыка ...)
- Матрица взаимодействий (user-item)

Что требуется?

#015

Что требуется?

- Предсказать оценку user-а U к товару i
- Дать персональные рекомендации
- Найти похожие товары

Оценки пользователей

Пользовательские оценки можно получить двумя способами:

явно (explicit ratings) – пользователь сам ставит рейтинг товару, оставляет отзыв, лайкает страницу

неявно (implicit ratings) – пользователь явно свое отношение не выражает, но можно сделать косвенный вывод из его действий: купил товар – значит он ему нравится, долго читал описание – значит есть интерес и т.п.

Признаки пользователей



#018

Признаки пользователей

- Пол
- Возраст
- Интересы (группы ok, vk)
- Посещенные страницы (title, description)
- Место нахождения
- Купленные/просмотренные товары
- Установленные приложения
- Любимые точки пользователей

И так далее..

Признаки item-ов



#020

Признаки item-ов

- наименование
- стоимость
- текстовое описание
- категория
- изображение item-а

Зависит от предметной области

Матрица взаимодействий

	фильм #1	фильм #2	фильм #3	фильм #4	фильм #5	фильм #6
Аня	4			5	2	3
Наташа		1	3		1	4
Светлана			4	2	2	
Юлиана	5		2			
Андрей			2		5	

Матрица взаимодействий

	фильм #1	фильм #2	фильм #3	фильм #4	фильм #5	фильм #6
Аня	4			5	2	3
Наташа		1	3		1	4
Светлана			4	2	2	
Юлиана	5		2			
Андрей			2	?	5	

Какие бывают рекомендательные системы



#024

Виды рекомендательных систем

- Кластеризация пользователей
- Модель совстречаемости
- Summary-based (неперсональные)
- Content-based (основанные на описании товара)
- Коллаборативная фильтрация
- Матричная факторизация

Кластеризация пользователей

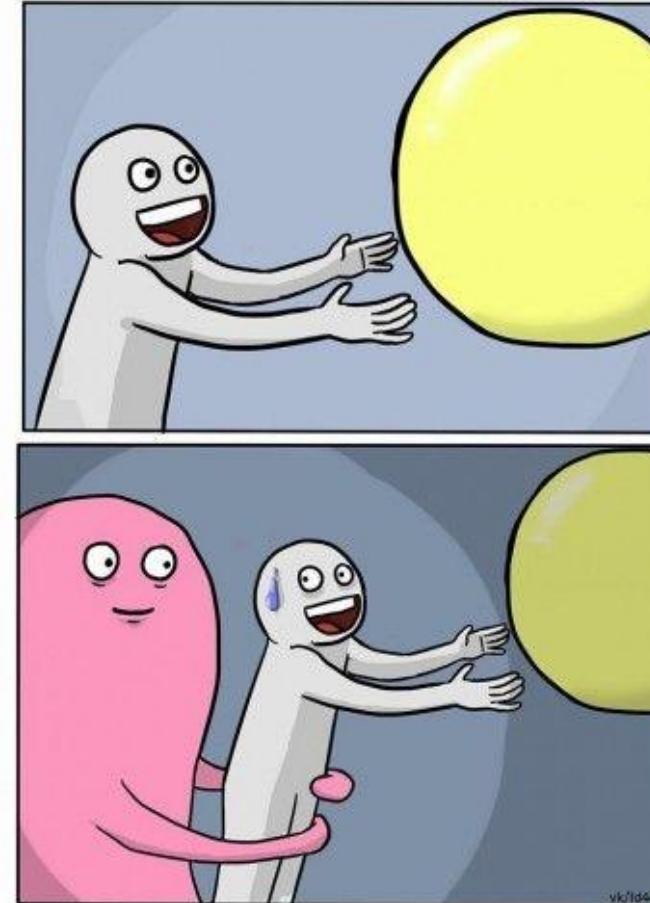
- Выберем меру схожести пользователей $sim(u, v)$ по истории их оценок
- Объединим пользователей в кластеры так, чтобы похожие пользователи попали в один кластер
- Оценку пользователя объекту будем предсказывать как среднюю оценку всего кластера по этому объекту

Кластеризация пользователей

$$\hat{r}_{ui} = \frac{1}{|F(u)|} \sum_{v \in F(u)} r_{vi}$$

Кластеризация пользователей

Проблемы:



#028

Кластеризация пользователей

Проблемы:

- Нечего рекомендовать **новым/нетипичным пользователям**
- Не учитывается специфика каждого пользователя. В каком-то смысле мы делим всех пользователей на какие-то классы (шаблоны).
- Если в кластере **никто не оценивал объект**, то предсказание сделать не получится

Модель совстречаемости (ассоциативные правила)

Похожими считаются товары, которые смотрят / кладут в корзину / покупают (нужное подчеркнуть) вместе

С ЭТИМ ТОВАРОМ ЧАСТО ПОКУПАЮТ



NOKIA X3-02

★★★★★ 10 800 руб.



SAMSUNG S3650 CORBY

★★★★★ 8 600 руб.



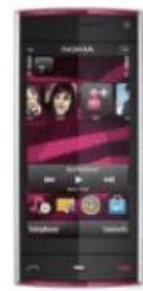
SONY ERICSSON XPERIA ARC

★★★★★ 38 800 руб.



SAMSUNG S7070 DIVA

★★★★★ 16 400 руб.



NOKIA X6 16GB

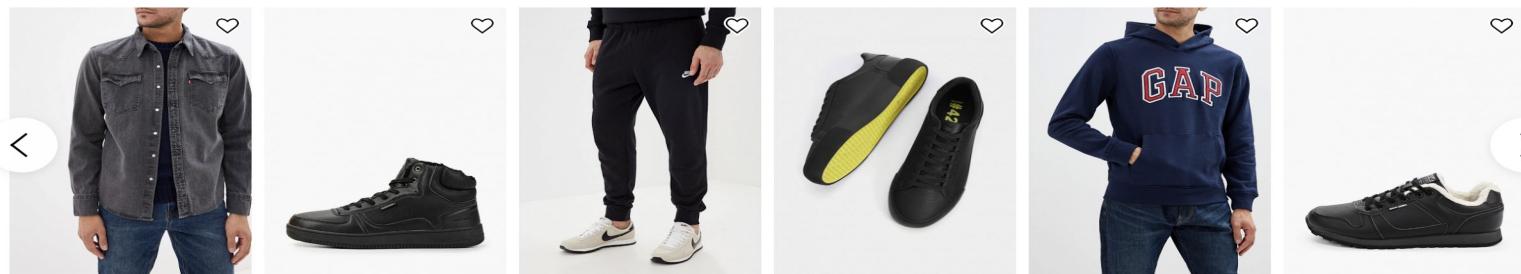
★★★★★ 25 200 руб.

Summary-based рекомендации

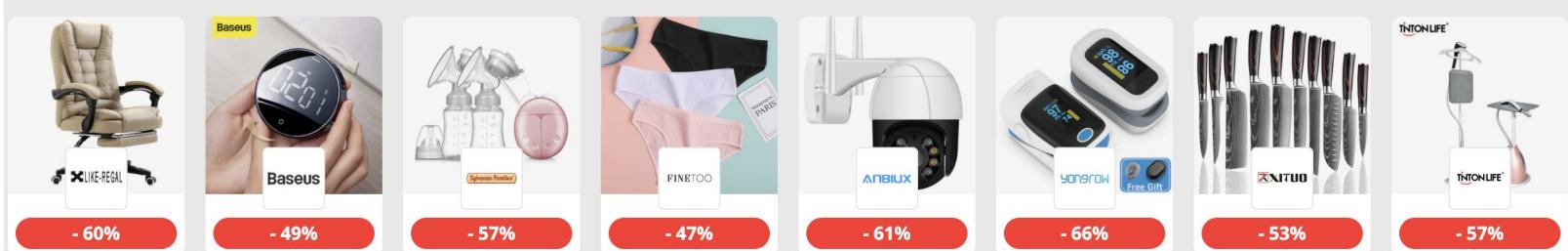
Неперсональные рекомендации – когда вам рекомендуют то же самое, что всем остальным.

Допускают таргетинг по региону или времени, но не учитывают ваши личные предпочтения.

ПОПУЛЯРНОЕ



Популярные бренды



Summary-based рекомендации

Можете ли привести пример неперсональных рекомендаций связанных с сезонностью?

Summary-based рекомендации

Можете ли привести пример неперсональных рекомендаций связанных с сезонностью?

Осенью чаще рекламируют средства от гриппа и т.д.

Под Новый Год больше рекламы Кока-Колы



Не Summary-based рекомендации

Персональные рекомендации используют всю доступную информацию о клиенте.

Рекомендуем вам



11.11 2 076,08 руб.



48 877,00 руб.



11.11 3 829,66 руб.



11.11 135,86 руб.



424,06 руб.



11.11 1 466,71 руб.

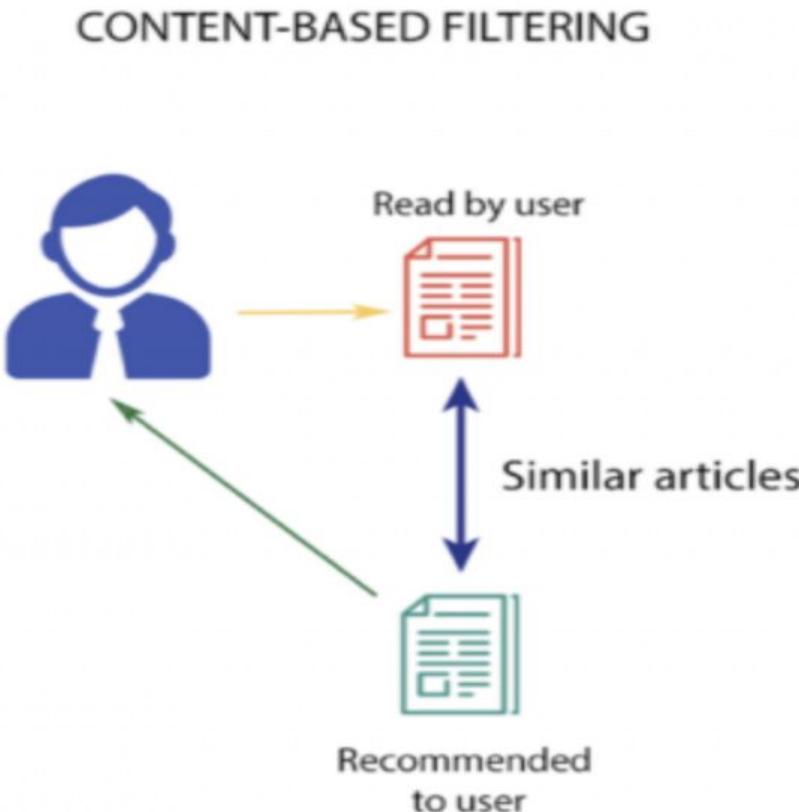
#034

Content-based рекомендации

Описание товара сопоставляется с интересами пользователя, полученными из его предыдущих оценок.

Купил утюг - теперь его предлагают на всех сайтах с Desktop, в телефоне, в объявлении в подъезде и даже во сне!

Content-based рекомендации



Если есть хорошие признаковые описания пользователей и объектов (и только они), тогда

$$\begin{aligned} u &\sim f_u \\ i &\sim f_i \end{aligned}$$

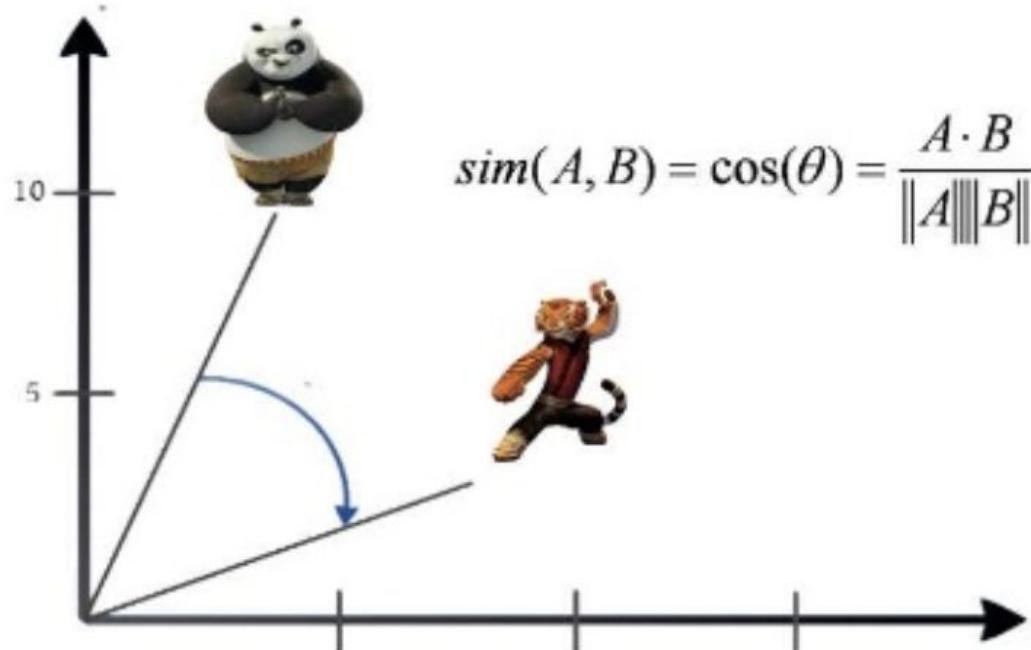
Можно решать как обычную задачу обучения с учителем

$$\{([f_u, f_i], r_{ui})\}$$

Цель: $u \rightarrow i_1, \dots, i_k : \hat{r}_{ui_1} \geq \hat{r}_{ui_2} \geq \dots$

Content-based рекомендации

Cosine Similarity



	утюг	увеличенное	отверстие	для	залива	воды	регулировка	температуры	режим	сухого	глажения	регулируемая	подача	пара
item1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
item2	0	0	0	1	0	1	0	0	0	0	0	0	0	0
item3	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Content-based рекомендации

Что использовать?

- TF-IDF
- Word2Vec
- Doc2Vec

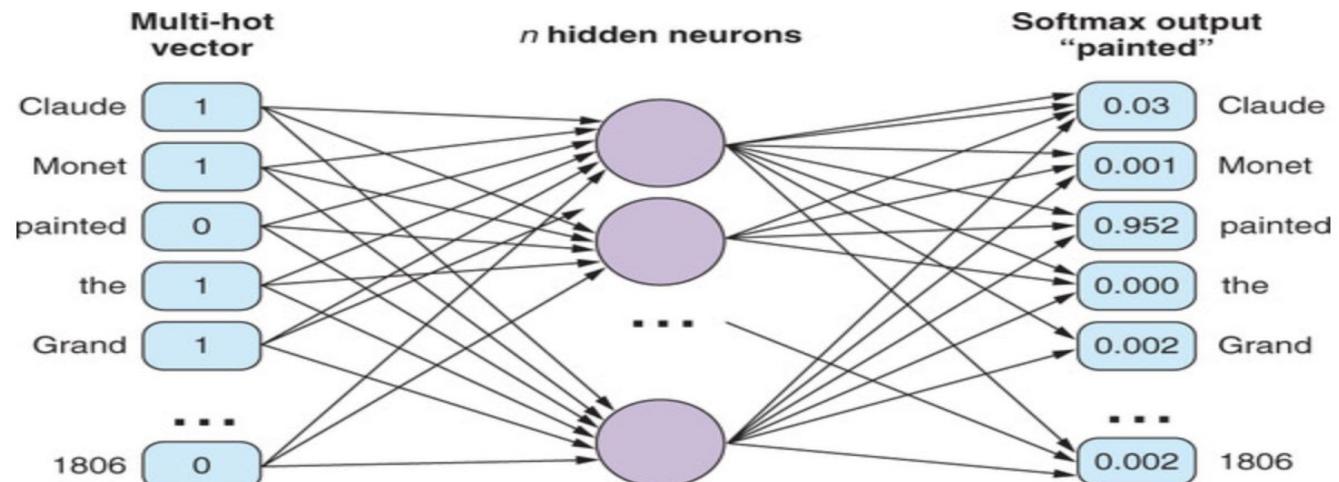
$$w_{i,j} = tf_{i,j} \times \log \left(\frac{N}{df_i} \right)$$

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

N = total number of documents

Сжатие размерности: SVD/ PCA



Content-based рекомендации

Преимущества	Недостатки
Независимость от данных других пользователей	Когда появляется новый пользователь с недостатком данных о его транзакциях, мы не можем качественно делать рекомендации.
Нет проблемы "холодного старта" для новых предметов, т.к. используя признаки предметов, мы можем легко находить похожие предметы	Формирование четких групп похожих продуктов может ограничить рекомендации других продуктов. Мы можем снова и снова рекомендовать лишь малое подмножество из всех продуктов
Результаты рекомендаций интерпретируемые	Если информация о продуктах ограничена, трудно различать предметы и группировать их. В результате качество рекомендаций будет низким

Вопрос

А есть ли еще способ использовать W2V для рекомендаций?

Вспомним, что такое W2V:

Word2Vec - это класс моделей нейронных сетей, которые изначально были представлены для изучения встраивания слов, которые очень полезны для задач обработки естественного языка.

Word2Vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он генерирует словарь корпуса, а затем вычисляет векторное представление слов, «обучаясь» на входных текстах.

Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие векторы

Как можно применить W2V для рекомендаций?

#042

Как можно применить W2V для рекомендаций?

Наши предложения – последовательность товаров, просмотренных пользователем за сессию (30 минут, 2 часа, неделя, 2 недели)

Слова – товар

Рекомендации – самые ближайшие товары к рассматриваемому по Word2Vec

Это |2| рекомендации



Как можно применить W2V для рекомендаций?

Можем сделать и персональные рекомендации

Усредня все эмбеддинги товаров просмотренных пользователем получаем вектор для пользователя.

Находим к нему ближайшие товары

W2V рекомендации

Эту технологию применили в своих движках рекомендаций такие компании, как [Airbnb](#), [Alibaba](#), [Spotify](#) и [Anghami](#).

Коллаборативная фильтрация

Если известна лишь статистика:

$$\{(u, i, r_{ui})\}$$

нет содержательных признаков!

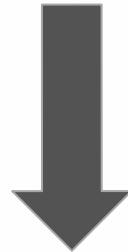
Решение на статистике поведения лучше, чем на описаниях!

Коллаборативная фильтрация



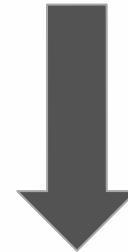
Коллаборативная фильтрация

Item-based



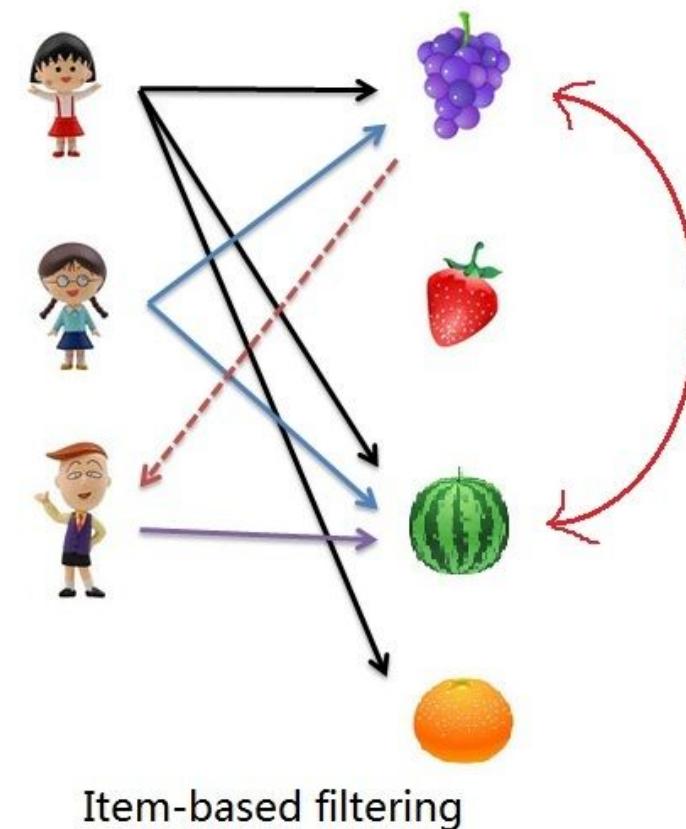
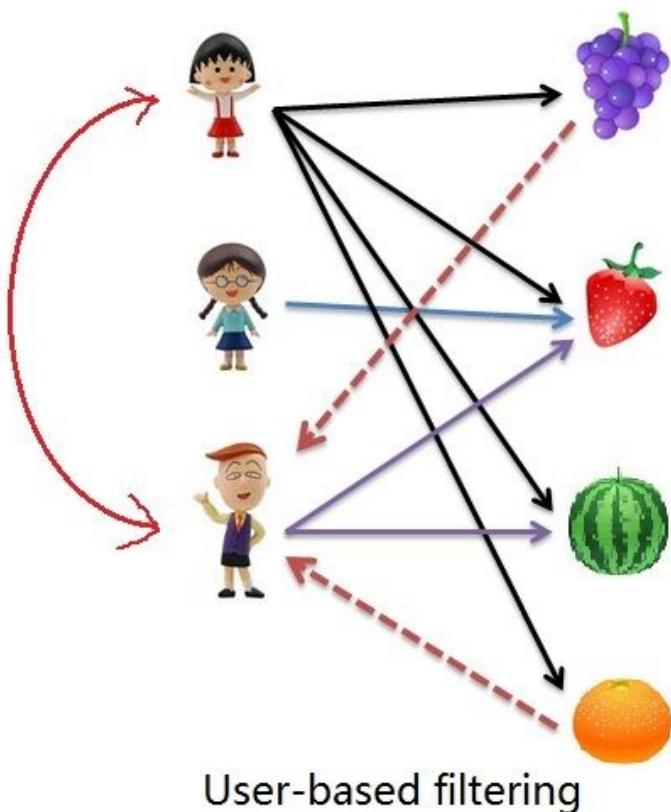
Предложи к твоим
товарам другие
связанные с ними
товары (чехол к
телефону, чипсы к
пиву и т.д.)

User-based



Предложи товары
или услуги,
которые есть у
твоих друзей
(похожих
пользователей)

Коллаборативная фильтрация



Коллаборативная фильтрация

Подход на основе памяти

Запоминаем матрицу interaction-ов, рекомендации составляются путем запроса данного пользователя к остальной части матрицы полезности.



Коллаборативная фильтрация

По пользователям (User-based)

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_v \text{sim}(u, v)(r_{vi} - \bar{r}_v)}{\sum_v \text{sim}(u, v)}$$

По товарам (Item-based)

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_j \text{sim}(i, j)(r_{uj} - \bar{r}_j)}{\sum_j \text{sim}(i, j)}$$

Коллаборативная фильтрация

Что лучше? item-based или user-based?

Коллаборативная фильтрация

Что лучше? item-based или user-based?

- Когда пользователей много (почти всегда), задача поиска ближайшего соседа становится плохо вычислимой

Коллаборативная фильтрация

Что лучше? item-based или user-based?

- Когда пользователей много (почти всегда), задача поиска ближайшего соседа становится плохо вычислимой
- Оценка близости товаров гораздо более точная, чем оценка близости пользователей

Коллаборативная фильтрация

Что лучше? item-based или user-based?

- Когда пользователей много (почти всегда), задача поиска ближайшего соседа становится плохо вычислимой
- Оценка близости товаров гораздо более точная, чем оценка близости пользователей
- В user-based варианте описания пользователей, как правило, сильно разрежены (товаров много, оценок мало). Проблема: сколько соседей не бери, список товаров, которые в итоге можно порекомендовать, получается очень небольшим

Коллаборативная фильтрация

Что лучше? item-based или user-based?

- Когда пользователей много (почти всегда), задача поиска ближайшего соседа становится плохо вычислимой
- Оценка близости товаров гораздо более точная, чем оценка близости пользователей
- В user-based варианте описания пользователей, как правило, сильно разрежены (товаров много, оценок мало). Проблема: сколько соседей не бери, список товаров, которые в итоге можно порекомендовать, получается очень небольшим

Коллаборативная фильтрация

Что лучше? item-based или user-based?

- Нечего рекомендовать новым / нетипичным пользователям. Для таких пользователей мы все еще не можем найти похожих.
- Предпочтения пользователя могут меняться со временем, но описание товаров штука гораздо более устойчивая

Коллаборативная фильтрация

Общие проблемы:

- Все перечисленные методы обладают следующими недостатками:
Проблема холодного старта.
- Плохие предсказания для новых/нетипичных пользователей/объектов.
- Тривиальность рекомендаций.
- Ресурсоемкость вычислений. Для того, чтобы делать предсказания нам нужно держать в памяти все оценки всех пользователей.

Коллаборативная фильтрация

Метрики схожести

- Косинусная мера

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

- Коэффициент корреляции Пирсона

$$\text{sim}(u, v) = \frac{\sum_i (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_i (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_i (r_{vi} - \bar{r}_v)^2}}$$

- Евклидово расстояние

$$d(a, b) = \sqrt{\sum_{k=1}^n (a_k - b_k)^2}$$

$$d(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- Коэффициент Жаккара

$$d(a, b) = \sum_{k=1}^n |a_k - b_k|$$

- Манхэттенское расстояние и т.д.

Коллаборативная фильтрация

Подход на основе модели

Одной из наиболее распространенных реализаций подхода на основе модели является **матричная факторизация**.

SVD (Singular Value Decomposition)

В теореме о сингулярном разложении утверждается, что у любой матрицы A размера $n \times m$ существует разложение в произведение трех матриц: U , Σ и V :

$$A_{n \times m} = U_{n \times n} \times \Sigma_{n \times m} \times V^T_{m \times m}$$

$$UU^T = I_n, \quad VV^T = I_m,$$

$$\Sigma = \text{diag}(\lambda_1, \dots, \lambda_{\min(n,m)}), \quad \lambda_1 \geq \dots \geq \lambda_{\min(n,m)} \geq 0$$

Усеченное SVD (Singular Value Decomposition)

$$\underset{n \times m}{A} = \underset{n \times n}{U} \times \underset{n \times m}{\Sigma} \times \underset{m \times m}{V^T}$$

$$\lambda_{d+1}, \dots, \lambda_{\min(n,m)} := 0$$

$$\underset{}{A'} = \underset{}{U'} \times \underset{}{\Sigma'} \times \underset{}{{V'}^T}$$

Усеченное SVD (Singular Value Decomposition)

$$\underset{n \times m}{A} = \underset{n \times n}{U} \times \underset{n \times m}{\Sigma} \times \underset{m \times m}{V^T}$$

$$\lambda_{d+1}, \dots, \lambda_{\min(n,m)} := 0$$

$$\underset{n \times m}{A'} = \underset{n \times d}{U'} \times \underset{d \times d}{\Sigma'} \times \underset{d \times m}{V'^T}$$

Усеченное SVD (Singular Value Decomposition)

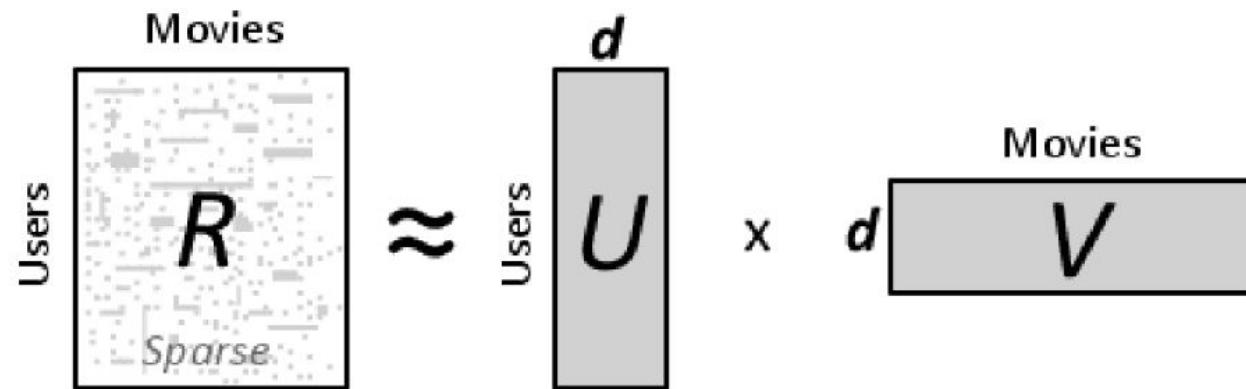
$$\lambda_{d+1}, \dots, \lambda_{\min(n,m)} := 0$$

$$\begin{matrix} \mathbf{A}' \\ n \times m \end{matrix} = \begin{matrix} \mathbf{U}' \\ n \times d \end{matrix} \times \begin{matrix} \Sigma' \\ d \times d \end{matrix} \times \begin{matrix} \mathbf{V}'^T \\ d \times m \end{matrix}$$

Полученная матрица \mathbf{A}' хорошо приближает исходную матрицу \mathbf{A} и, более того, является **наилучшим** низкоранговым приближением с точки зрения средне-квадратичного отклонения.

Матричная факторизация

Как же использовать все это для рекомендаций? У нас была матрица, мы разложили ее в произведение трех матриц. При чем разложили не точно, а приблизительно. Упростим все немного, обозначив произведение первых двух матриц за одну матрицу:



$$\hat{r}_{ui} = \langle p_u, q_i \rangle$$

Скрытые представления пользователей и товаров!

Матричная факторизация

$$\hat{r}_{ui}(\Theta) = \mathbf{p}_u^T \mathbf{q}_i,$$

$$\Theta = \{\mathbf{p}_u, \mathbf{q}_i \mid u \in U, i \in I\}$$

Нужно: подобрать параметры так, чтобы на тех оценках, которые у нас уже есть, ошибка была как можно меньше

$$\mathbf{E}_{(u,i)} (\hat{r}_{ui}(\Theta) - r_{ui})^2 \rightarrow \min_{\Theta}$$

Чего-то не хватает...

Матричная факторизация

$$\underbrace{\sum_{(u,i) \in \mathcal{D}} (\hat{r}_{ui}(\Theta) - r_{ui})^2}_{\text{качество на обучающей выборке}} + \lambda \underbrace{\sum_{\theta \in \Theta} \theta^2}_{\text{регуляризация}} \rightarrow \min_{\Theta}$$

Как найти оптимальные параметры?

Матричная факторизация

$$\underbrace{\sum_{(u,i) \in \mathcal{D}} (\hat{r}_{ui}(\Theta) - r_{ui})^2}_{\text{качество на обучающей выборке}} + \underbrace{\lambda \sum_{\theta \in \Theta} \theta^2}_{\text{регуляризация}} \rightarrow \min_{\Theta}$$

$$J(\Theta) = \sum_{(u,i) \in \mathcal{D}} (p_u^T q_i - r_{ui})^2 + \lambda \left(\sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 \right)$$

Матричная факторизация

Как найти минимум функции, зависящей от большого количества переменных?

Правильно, нам потребуется **градиент**

$$\nabla J(\Theta) = \left(\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \dots, \frac{\partial J}{\partial \theta_n} \right)^T$$

Самый известный метод оптимизации функций — **градиентный спуск**

Матричная факторизация

Для чего нужен градиент?

Градиент в какой-нибудь конкретной точке — это такой вектор, направленный в ту сторону, куда больше всего растет наша функция.

Соответственно, чтобы минимизировать наш функционал нужно..

Матричная факторизация

Для чего нужен градиент?

Градиент в какой-нибудь конкретной точке — это такой вектор, направленный в ту сторону, куда больше всего растет наша функция.

Соответственно, чтобы минимизировать наш функционал нужно.. двигаться в сторону антиградиента

$$\Theta_{t+1} = \Theta_t - \eta \nabla J(\Theta)$$

ALS факторизация

В этом подходе оценка r_{ui} пользователя u , поставленная фильму i , ищется как скалярное произведение векторов p_u и q_i в некотором пространстве \mathbb{R}^K латентных признаков:

$$\hat{r}_{ui} = p_u^T q_i$$

Иными словами, модель находит пространство признаков, в котором мы описываем и фильмы и пользователей и в котором рейтинг является мерой близости между фильмами и пользователями.

Для настройки модели будем минимизировать следующую ошибку:

$$\sum_{(u,i,r_{ui})} (r_{ui} - p_u^T q_i)^2 + \lambda_p p_u^T p_u + \lambda_q q_i^T q_i,$$

где суммирование ведется по всем тройкам (u, i, r_{ui}) выборки, слагаемые с λ_p и λ_q добавлены для регуляризации.



Это все замечательно, но что в реальности?

#073

Реальность

Можно выделить два подхода к формированию выдачи:

1. Item2Item
2. User2Item



Плюсы и минусы Item2Item

+ Быстро



Плюсы и минусы Item2Item

- + Быстро
- + Дешево (совстречаемость, можно пересчитывать не очень часто)



Плюсы и минусы Item2Item

- + Быстро
- + Дешево (совстречаемость, можно пересчитывать не очень часто)
- + Понятно, интерпретируемо

Плюсы и минусы Item2Item

- + Быстро
- + Дешево (совстречаемость, можно пересчитывать не очень часто)
- + Понятно, интерпретируемо
- Не учитывает особенности пользователя (кроме истории посещений).

Плюсы и минусы Item2Item

- + Быстро
- + Дешево (совстречаемость, можно пересчитывать не очень часто)
- + Понятно, интерпретируемо
- Не учитывает особенности пользователя (кроме истории посещений).
- Холодный старт

Плюсы и минусы User2Item

- + Учитывает особенности пользователя

Плюсы и минусы User2Item

- + Учитывает особенности пользователя
- + Решается проблема с холодным стартом (если хоть что-то знаем о пользователе уже можем скорее всего что-то адекватное рекомендовать)

Плюсы и минусы User2Item

- + Учитывает особенности пользователя
- + Решается проблема с холодным стартом (если хоть что-то знаем о пользователе уже можем скорее всего что-то адекватное рекомендовать)
- Нужно часто пересчитывать, обновлять модель/вектора (даже один клик несет в себе очень много информации)

Плюсы и минусы User2Item

- + Учитывает особенности пользователя
- + Решается проблема с холодным стартом (если хоть что-то знаем о пользователе уже можем скорее всего что-то адекватное рекомендовать)
- Нужно часто пересчитывать, обновлять модель/вектора (даже один клик несет в себе очень много информации)
- Дорого считать (учитывая предыдущий пункт нужно хранить где-то вектор юзера и товаров, индексы, проводить поиск и ранжирование)

Плюсы и минусы User2Item

- + Учитывает особенности пользователя
- + Решается проблема с холодным стартом (если хоть что-то знаем о пользователе уже можем скорее всего что-то адекватное рекомендовать)
- Нужно часто пересчитывать, обновлять модель/вектора (даже один клик несет в себе очень много информации)
- Дорого считать (учитывая предыдущий пункт нужно хранить где-то вектор юзера и товаров, индексы, проводить поиск и ранжирование)
- +/- Можно проводить расчёт аналогично Item2Item (раз в день), но из-за этого выдача будет статична, после клика по новым объектам не будет меняться

Плюсы и минусы User2Item

- + Учитывает особенности пользователя
- + Решается проблема с холодным стартом (если хоть что-то знаем о пользователе уже можем скорее всего что-то адекватное рекомендовать)
- Нужно часто пересчитывать, обновлять модель/вектора (даже один клик несет в себе очень много информации)
- Дорого считать (учитывая предыдущий пункт нужно хранить где-то вектор юзера и товаров, индексы, проводить поиск и ранжирование)
- +/- Можно проводить расчёт аналогично Item2Item (раз в день), но из-за этого выдача будет статична, после клика по новым объектам не будет меняться
- Интерпретировать скорее всего будет сложнее

Реальность

Стандартный пайплайн:

1. Подготовка рекомендаций (оффлайн, в ходу)

- Рассчет дневных статистик, фичей.
- Агрегирование, подбор кандидатов.
- Фильтрация/дополнительное ранжирование.

2. Ответ на запрос, формирование выдачи (онлайн)

- Разбор запроса, поход за рекомендациями в базу данных.
- Досортировка (по какому-то полю), фильтрация.
- Походы в базу данных за инфой для баннера, проверка.
- Запасные тактики (на случай, если с рекомендациями что-то не так).

Реальность

Мониторинг:

1. числа отрисованных/не отрисованных баннеров,
2. времени ответа (среднего, максимального),
3. количества походов в базу данных,
4. метрики по товаром (тип подбора (рекомендация, топ и тп), достаточно ли информации),
5. продуктовые метрики в разрезах тестов и ключевых магазинов,
6. рекомендаций и товаров (доля активных, покрытия и тп),
7. заливки в базу данных рекомендаций, топа, инфы о продуктах.

На некоторые ключевые показатели стоят алерты в Slack / Telegram / sms.

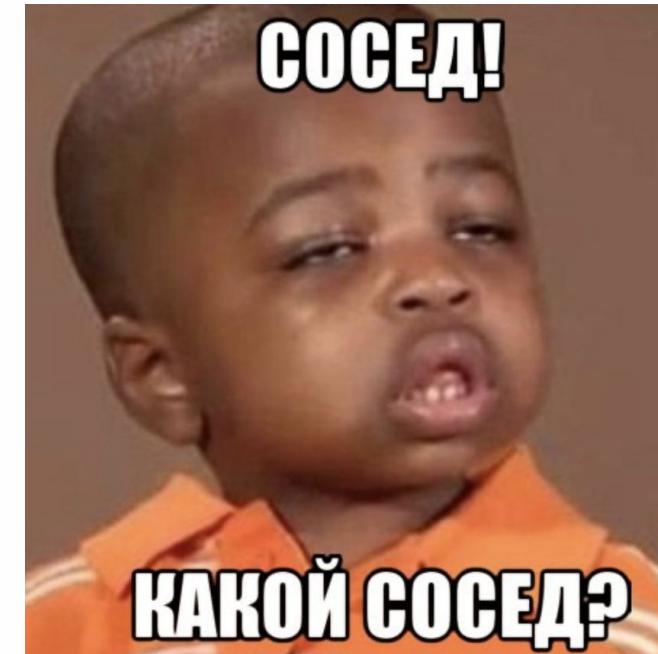
Проблемы рекомендательных систем

При большом числе user-ов или item-ов
поиск ближайших сущностей
нецелесообразно осуществлять полным
перебором.

Что делать?

Использовать приближенные методы
поиска ближайших соседей:

HNSW, ANNOY, LSH



Желаемые свойства рекомендаций

Разнообразие (diversity) ~ непохожие на другие товары из списка

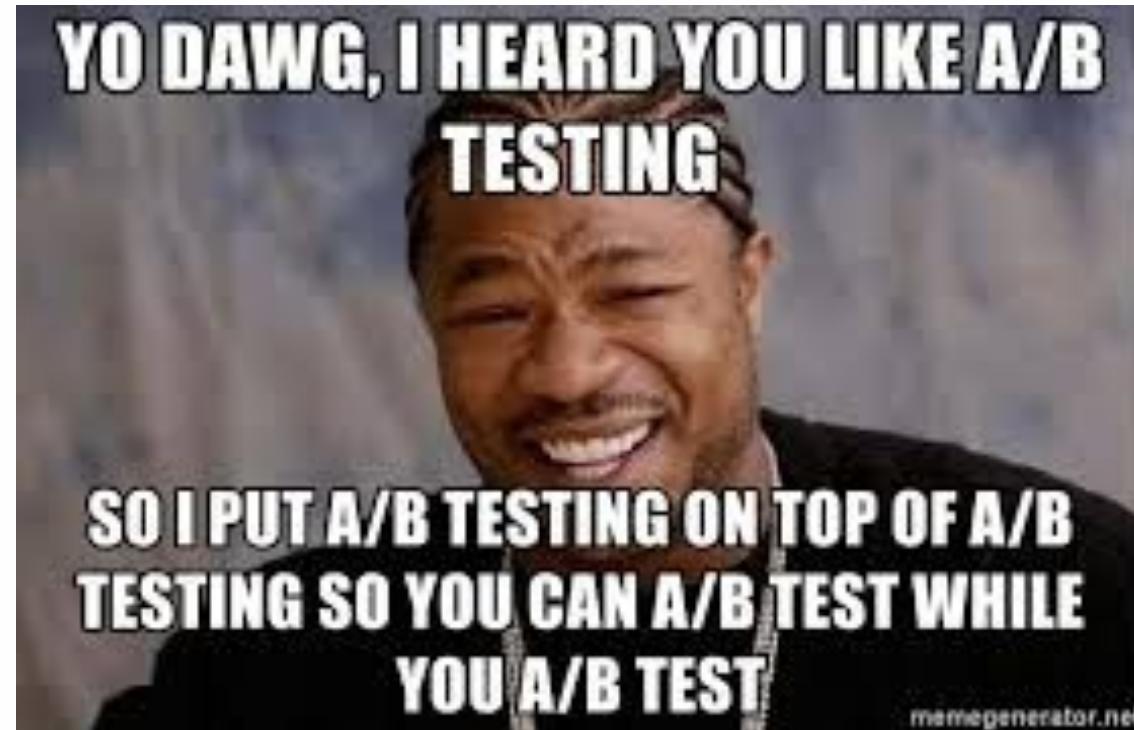
Плохо: к ноутбуку только ноутбуки того же производителя

Новизна (novelty) ~ для пользователя Плохо: каждый день одно и то же

Доверие ~ обосновать рекомендацию «с товаром покупают», «скидка за комплект», ...

Оценка качества рекомендательных систем

- offline-тестирование с помощью ретротестов
- A/B - тестирование



Метрики

Случай непрерывных оценок

Название	Формула	Описание
MAE (Mean Absolute Error)	$E(P - R)$	Среднее абсолютное отклонение
MSE (Mean Squared Error)	$E(P - R ^2)$	Среднеквадратичная ошибка
RMSE (Root Mean Squared Error)	$\sqrt{E(P - R ^2)}$	Корень из среднеквадратичной ошибки

Метрики

Случай дискретных оценок (бинарные)

Название	Формула	Описание
Precision	$\frac{TP}{TP+FP}$	Доля рекомендаций, понравившихся пользователю
Recall	$\frac{TP}{TP+FN}$	Доля интересных пользователю товаров, которая показана
F1-Measure	$\frac{2PR}{P+R}$	Среднее гармоническое метрик Precision и Recall. Полезно, когда заранее невозможно сказать, какая из метрик важнее
ROC AUC		Насколько высока концентрация интересных товаров в начале списка рекомендаций
Precision@N		Метрика Precision, посчитанная на Top-N записях
Recall@N		Метрика Recall, посчитанная на Top-N записях
AverageP		Среднее значение Precision на всем списке рекомендаций

Метрики

recommender system precision:

$$P = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of items we recommended}}$$

recommender system recall:

$$r = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of all the possible relevant items}}$$

P@k и **R@k** – это просто precision и recall, рассчитанные с учетом только подмножества рекомендаций от ранга 1 до k .

$$\text{AP@N} = \frac{1}{m} \sum_{k=1}^N (\text{P}(k) \text{ if } k^{\text{th}} \text{ item was relevant}) = \frac{1}{m} \sum_{k=1}^N P(k) \cdot \text{rel}(k)$$

rel (k) – это просто индикатор (0/1), который сообщает нам, был ли релевантен этот k -й элемент, а **P (k)** – точность @ k

Метрики

В пользовательской сессии были показаны рекомендации товаров А, Б и В.
Куплены при этом были товары Б и Д.

Знатоки, внимание вопрос:

- Чему равен R@3
для рекомендаций
в этом примере?
- А P@3?



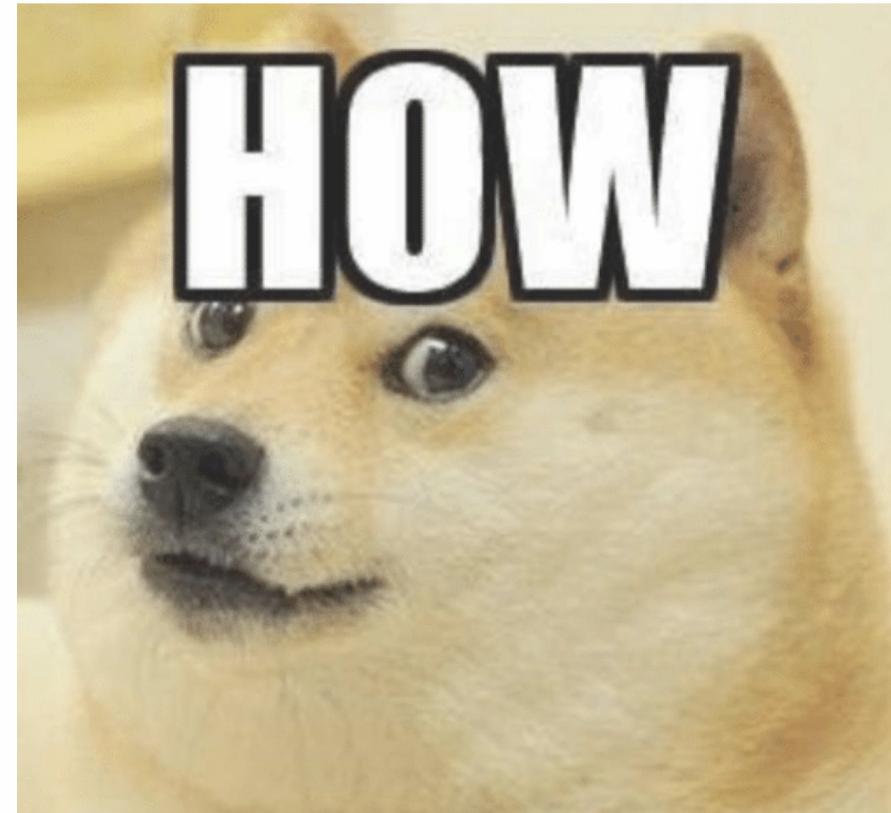
Метрики

Можно придумывать свои метрики. Как?

Пример:

Есть item2item товарные рекомендации. Для каждого товара имеется 10 похожих.

Как оценить их качество оффлайн?



Пример: «Условный» precision

Имеем 10 рекомендаций к каждому товару. Фиксируем 1 товар.

Получаем следующую метрику:

$$pr(i) = \frac{1}{10} \frac{1}{|I_i|} \sum_{u \in I_i} |I_u \cap rec_i|,$$

где rec_i — рекомендации для товара i , I_u — множество товаров посещенных пользователем u , I_i — множество пользователей, имеющих взаимодействие с товаром i

Вопрос

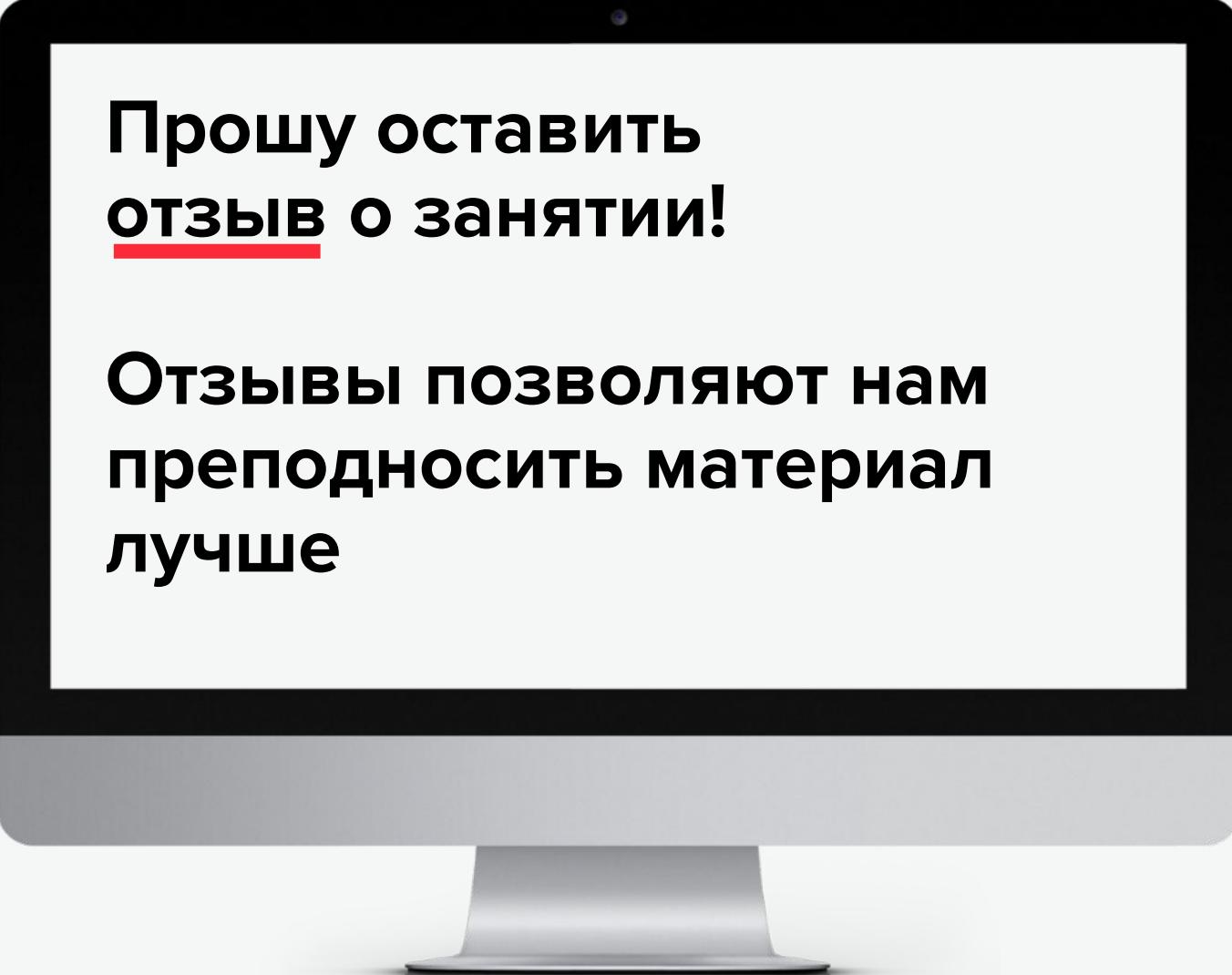
Какие из перечисленных ниже блоков соответствуют item2item рекомендациям:

- Блок "похожие товары" на странице товара
- Персональные рекомендации пользователю из всего каталога на основе его интересов
- Сопутствующие товары к данному товару
- Просмотренные пользователем товары
- Самые популярные товары
- Аксессуары к данному товару

Вопрос

Какие из перечисленных ниже блоков соответствуют item2item рекомендациям:

- Блок "похожие товары" на странице товара
- Персональные рекомендации пользователю из всего каталога на основе его интересов
- Сопутствующие товары к данному товару
- Просмотренные пользователем товары
- Самые популярные товары
- Аксессуары к данному товару



**Прошу оставить
отзыв о занятии!**

**Отзывы позволяют нам
преподносить материал
лучше**

**СПАСИБО
ЗА ВНИМАНИЕ**

