ТЕХНОСФЕРА

Лекция №8

Введение в машинное обучение

Журавлёв
Вадим

Введение в машинное обучение. Лекция 8

Отмечаемся и оставляем отзывы.

Спасибо!

Содержание лекции

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

Постановка задачи

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

Зачем нужны?

Рекомендательные системы

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

Предмет рекомендации

Что рекомендовать?

Предмет рекомендации

Товары	Книги Фильмы Музыка Игры Приложения
Контент	Новости Сайты Статьи
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

Предсказать:

- 2 800 000 оценок (float)

NETFLIX

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?

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

Требуется

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

Признаки пользователей

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

Признаки item-ов

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

Зависит от предметной области

Матрица взаимодействий (user-item)

	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 (user-item)

	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	

Рекомендательные системы

Какие же они бывают?

Рекомендательные системы

- 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

Проблемы:

Кластеризация пользователей

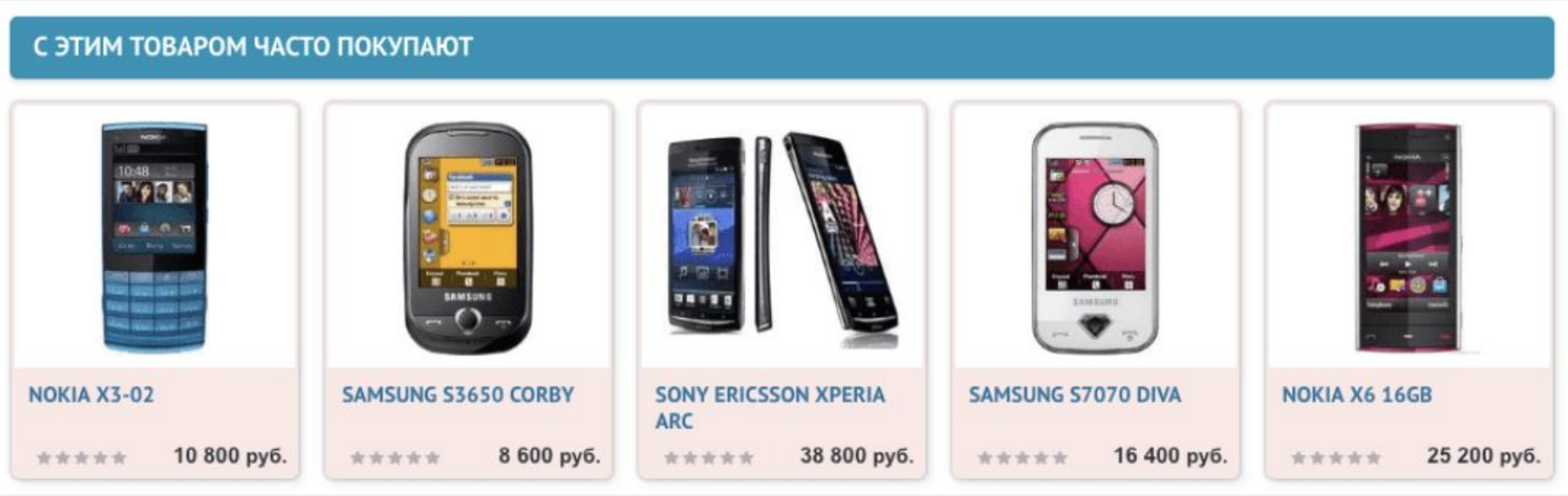
Проблемы:

- Нечего рекомендовать **новым/нетипичным пользователям**

- Если в кластере **никто не оценивал объект**, то предсказание сделать не получится

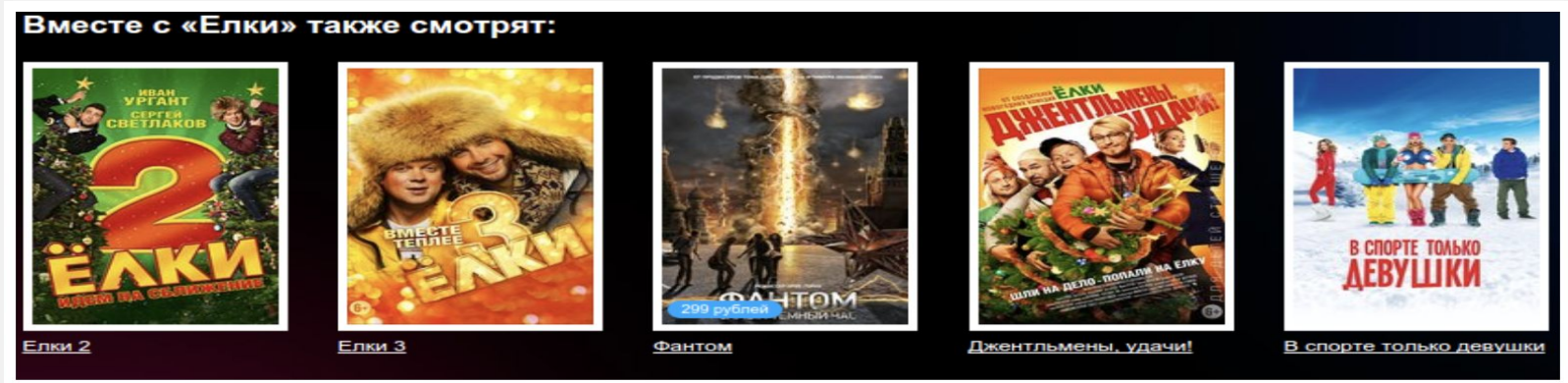
Модель совстречаемости (ассоциативные правила)

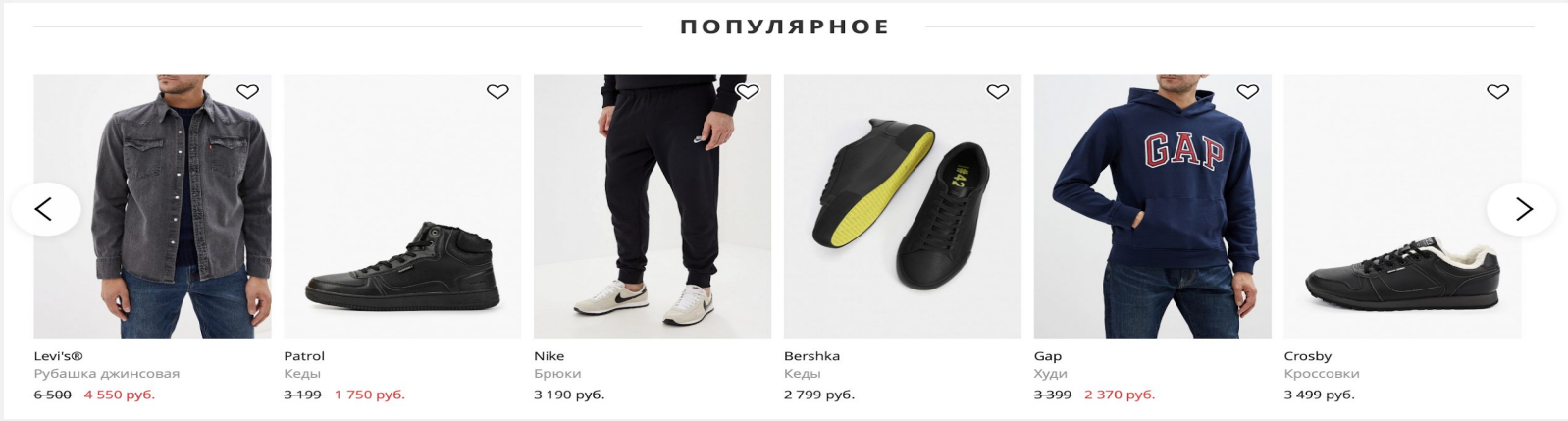
Похожими считаются товары, которые смотрят / кладут в корзину / покупают (нужное подчеркнуть) вместе.



Summary-based рекомендации

Неперсональные рекомендации – когда всем пользователям рекомендуется одно и тоже (например, топ товары).





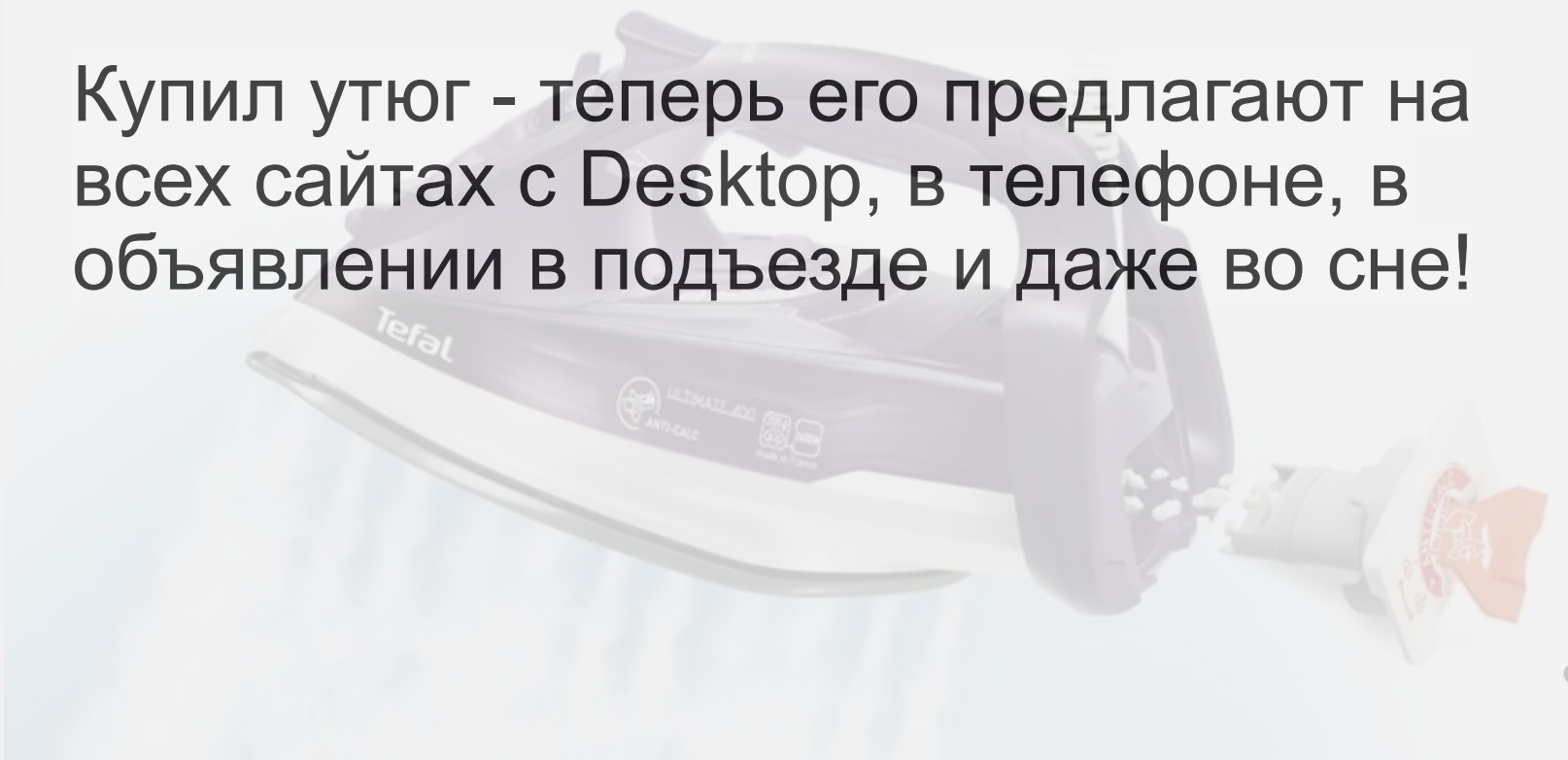
Не Summary-based рекомендации

Персональные рекомендации используют всю доступную информацию о клиенте.

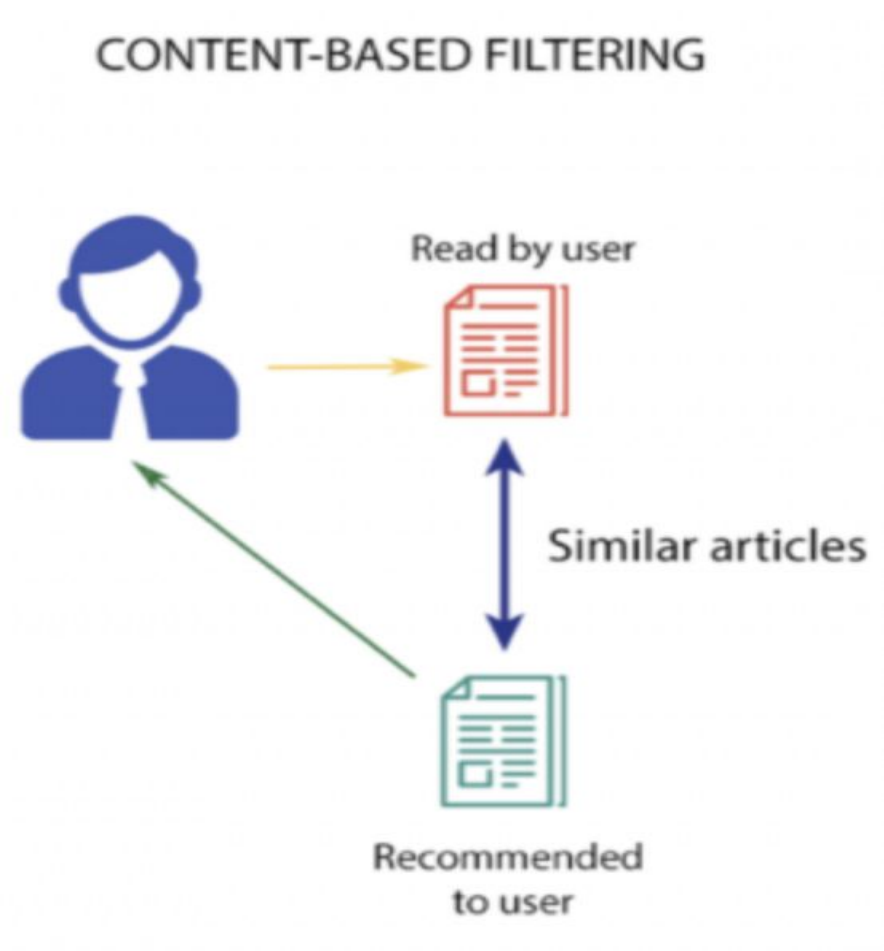
Content-based рекомендации

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

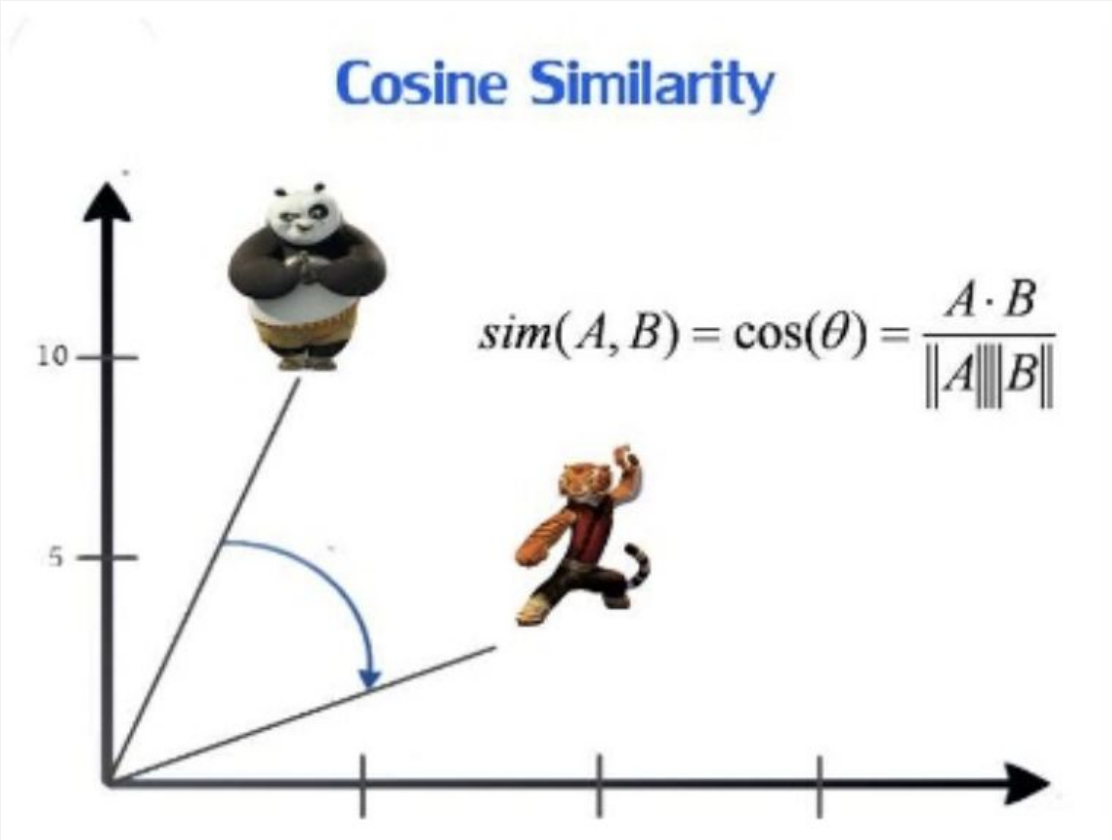
$$u \sim f_u$$
$$i \sim f_i$$

Можно решать как обычную задачу обучения с учителем

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

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

Content-based рекомендации



	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

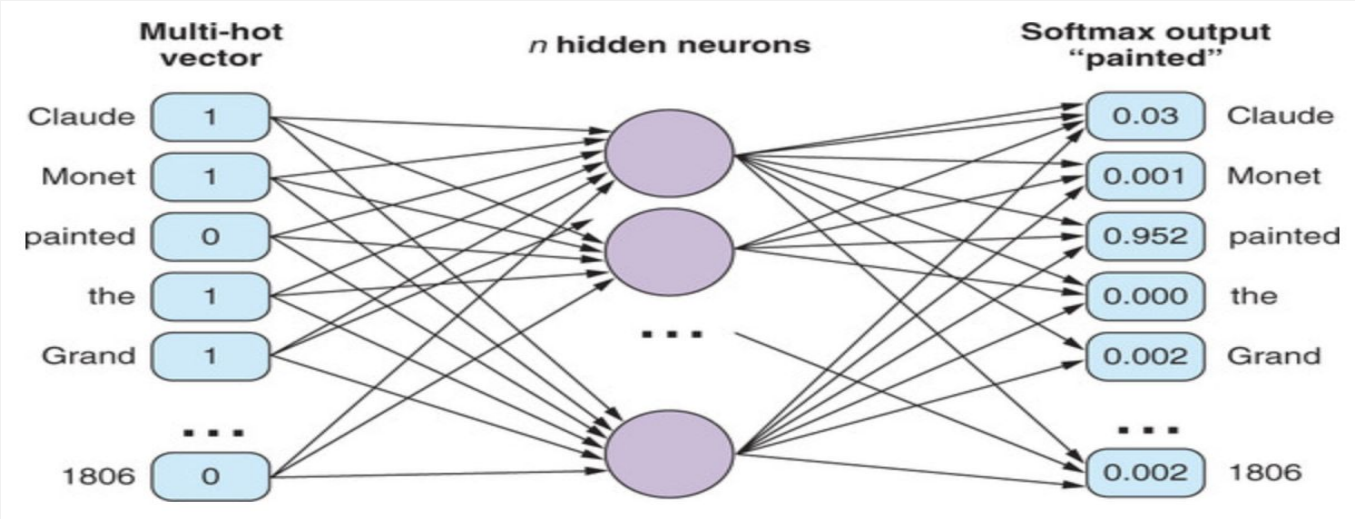
Что использовать?

- TF-IDF
- Word2Vec
- Doc2Vec

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

tf_{ij} = 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 интерпретируемы	Если информация о продуктах ограниченна, трудно различать предметы и группировать их. В результате качество рекомендаций будет низким

Коллаборативная фильтрация

Если известна лишь статистика:

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

нет содержательных признаков!

Решение на статистике поведения лучше, чем на описаниях!

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

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?

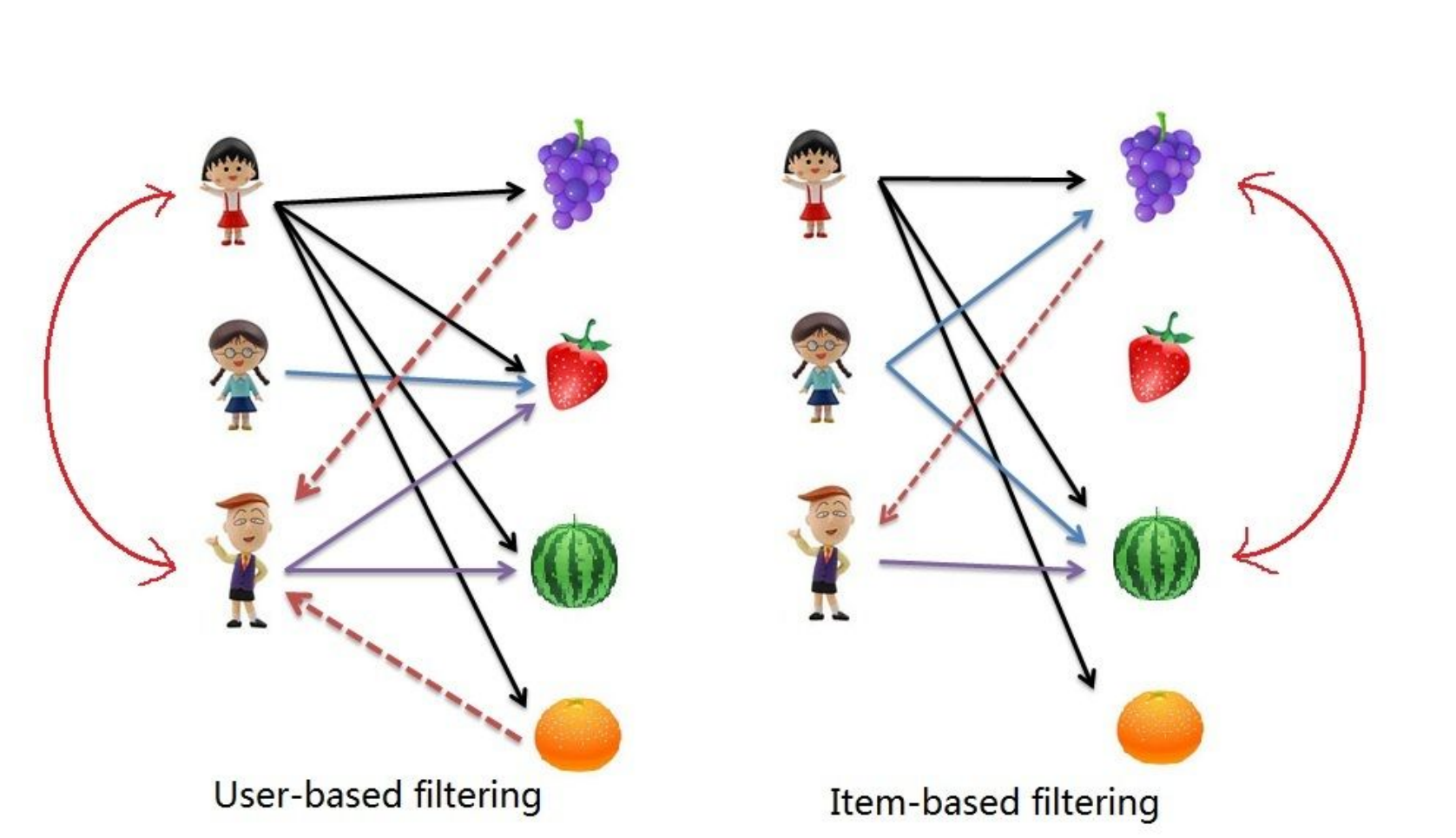
- Когда пользователей много (почти всегда), задача поиска ближайшего соседа становится плохо вычислимой;
- Оценка близости товаров гораздо более точная, чем оценка близости пользователей;
- В user-based варианте описания пользователей, как правило, сильно разрежены (товаров много, оценок мало). Проблема: 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;
- Предпочтения пользователя могут меняться со временем, но описание товаров штука гораздо более устойчивая.

Коллаборативная фильтрация



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

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

- Коэффициент Жаккара

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

- Манхэттенское расстояние и т.д.

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

Коллаборативная фильтрация

Коллаборативная фильтрация

Подход на основе памяти

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

SVD (Singular Value Decomposition)

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

$$\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}_n, \quad \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_m,$$

$$\boldsymbol{\Sigma} = \operatorname{diag}\left(\lambda_1, \ldots, \lambda_{\min(n,m)}\right), \quad \lambda_1 \geqslant \ldots \geqslant \lambda_{\min(n,m)} \geqslant 0$$

SVD (Singular Value Decomposition)

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

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

SVD (Singular Value Decomposition)

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

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

SVD (Singular Value Decomposition)

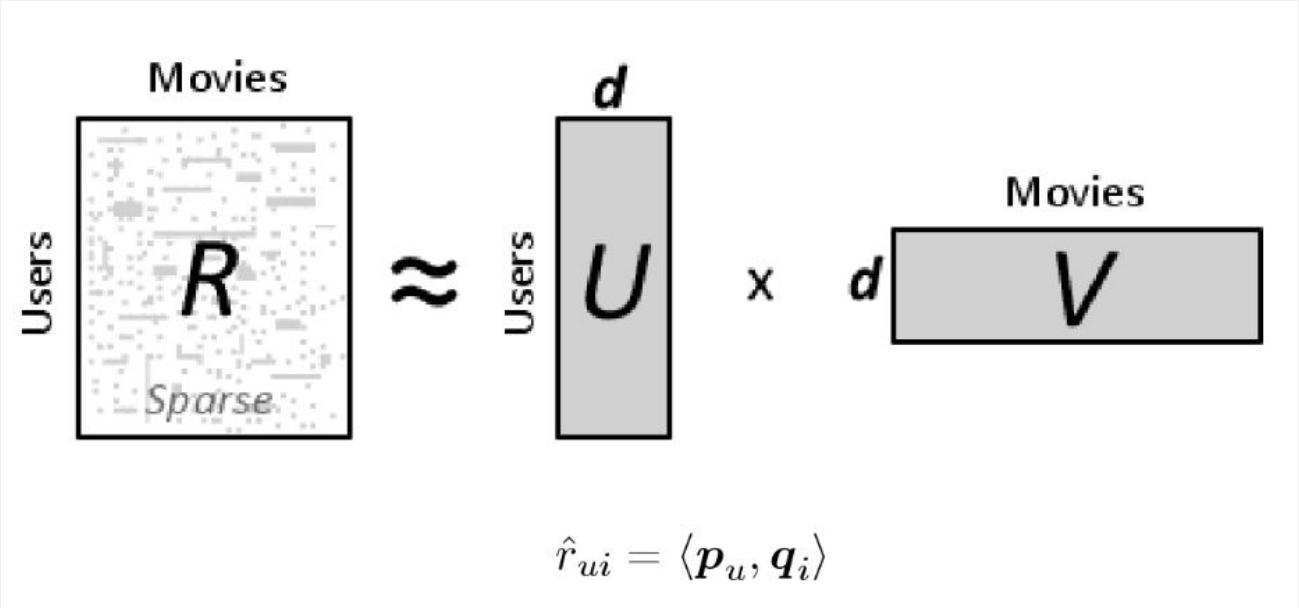
$$\lambda_{d+1}, \ldots, \lambda_{\min(n,m)} := 0$$

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

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

Коллаборативная фильтрация
Подход на основе памяти

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



Скрытые представления пользователей и товаров!

Матричная факторизация

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

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

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

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

Чего-то не хватает...

Матричная факторизация

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

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

Матричная факторизация

Как найти минимум функции, зависящей от большого количества переменных?

Правильно, нам потребуется **градиент**

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

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?

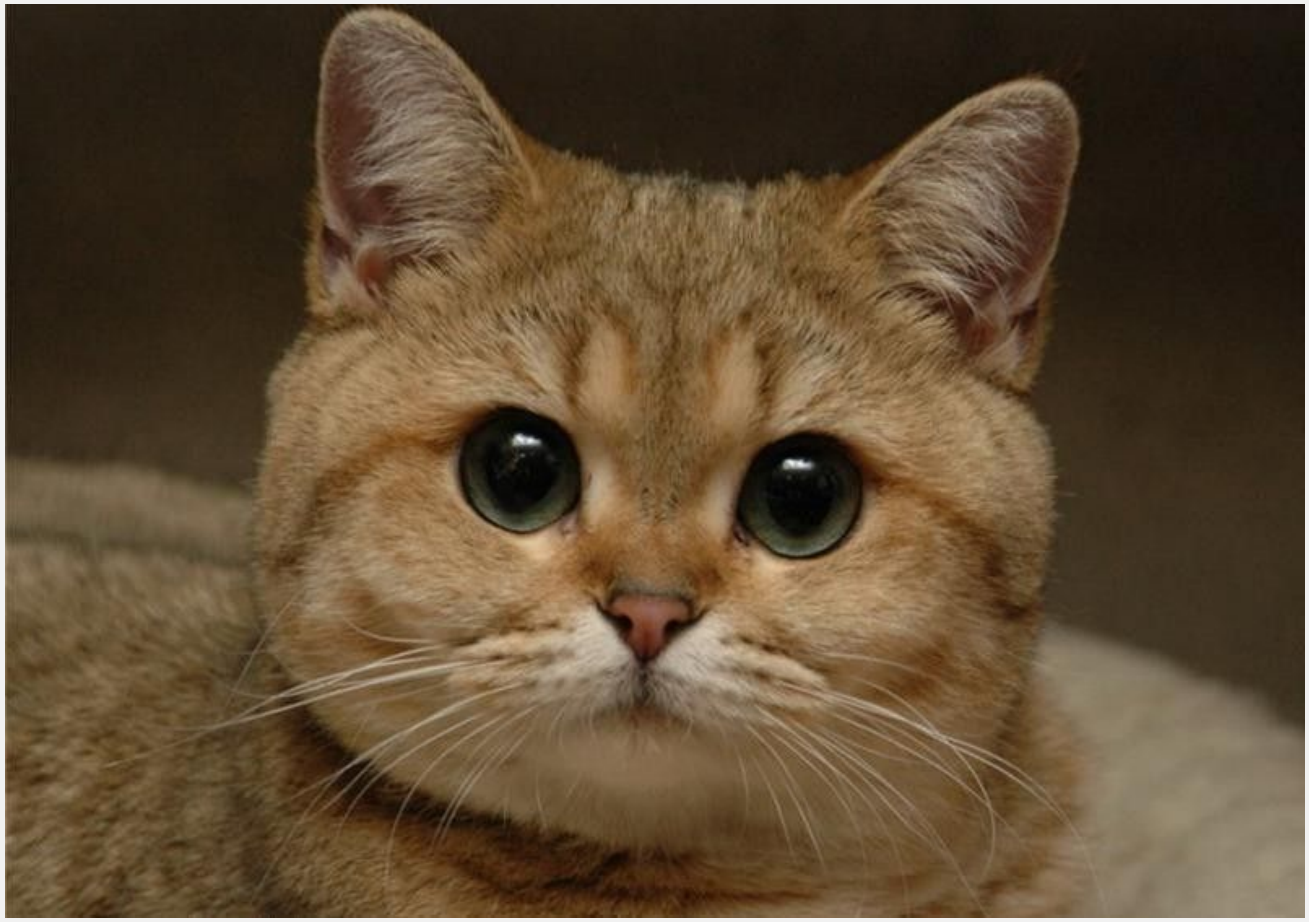
Градиент в какой-нибудь конкретной точке — это такой вектор, направленный в ту сторону, куда больше всего растет наша функция.

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

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

Примеры!

Пример в jupyter notebook



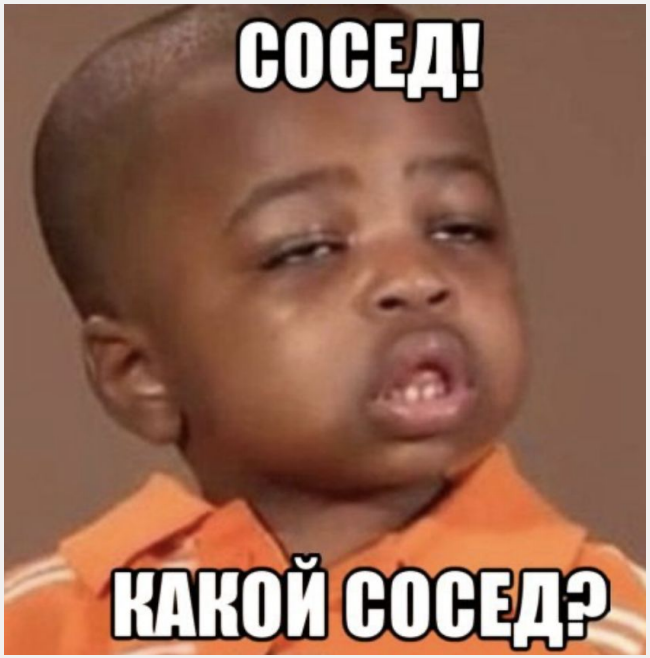
Проблемы рекомендательных систем

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 рекомендация

Разнообразие (diversity) ~ непохожие на другие товары из списка

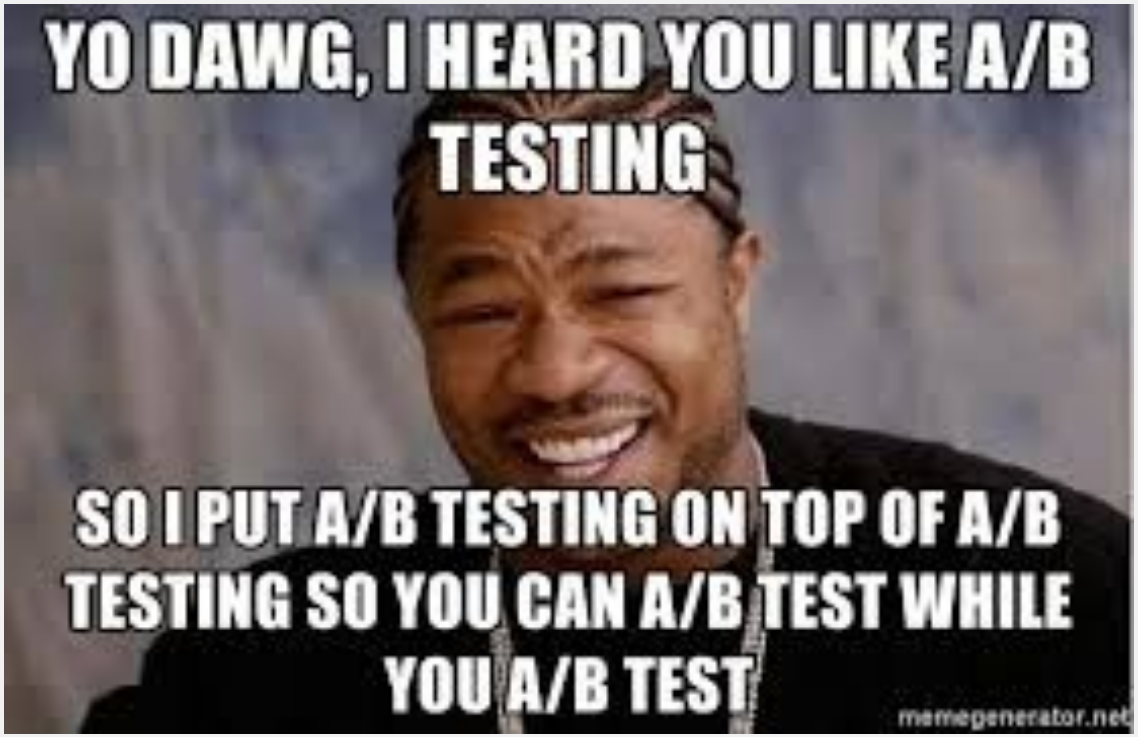
Плохо: к ноутбуку только ноутбуки того же производителя

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

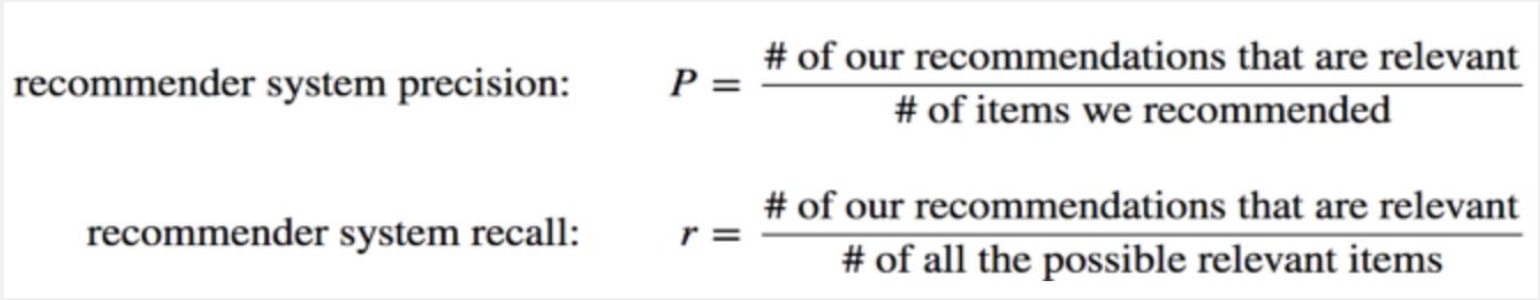
Название	Формула	Описание
MAE (Mean Absolute Error)	$E(\|P - R\|)$	Среднее абсолютное отклонение
MSE (Mean Squared Error)	$E(\|P - R\|^2)$	Среднеквадратичная ошибка
RMSE (Root Mean Squared Error)	$\sqrt{E(\|P - R\|^2)}$	Корень из среднеквадратичной ошибки

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 n.

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

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:

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?

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?

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

ТЕХНОСФЕРА

Спасибо за внимание!

Журавлёв Вадим

v.zhuravlyov@corp.mail.ru