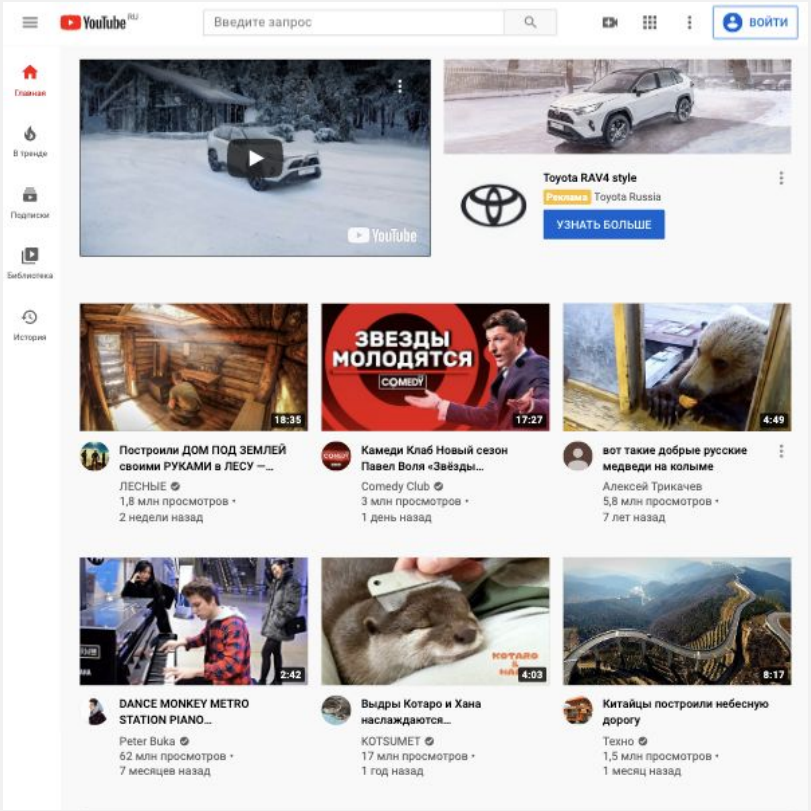
ТЕХНОСФЕРА

Лекция 6

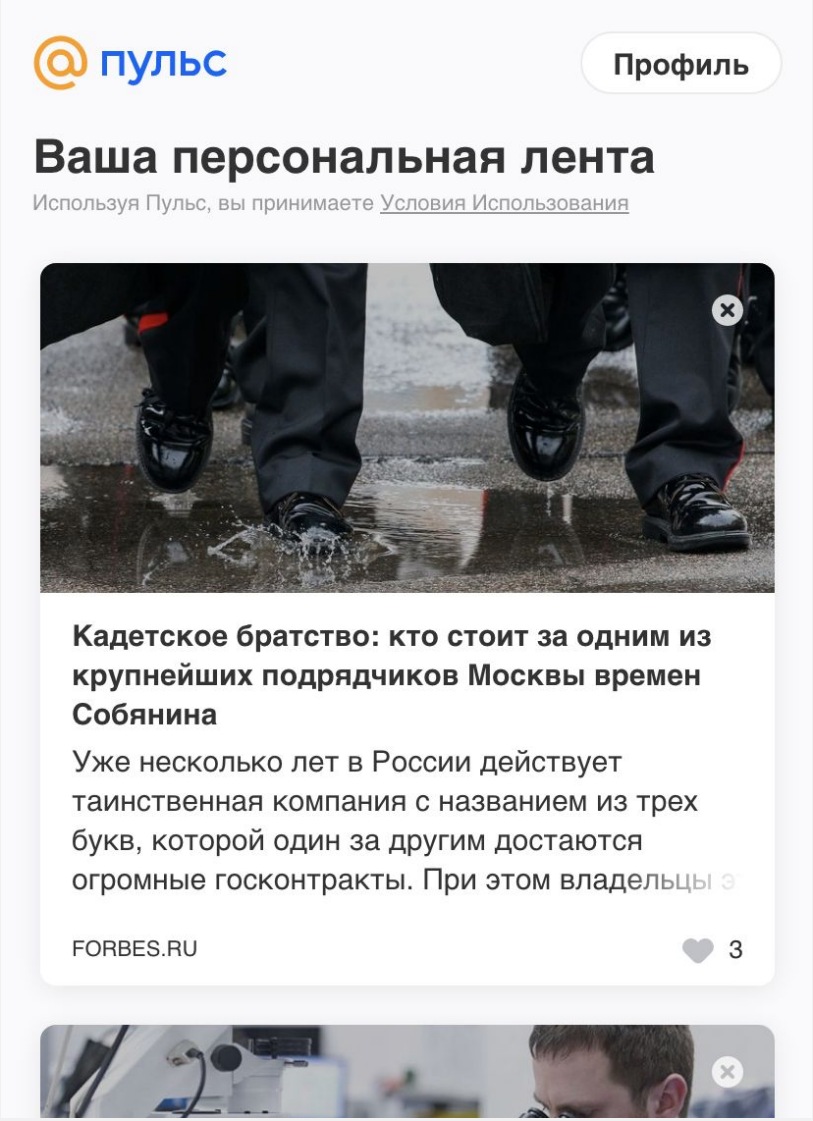
Рекомендательные системы

Чепарухин Сергей
Data Scientist@Mail.Ru

Что это?


YouTube RU
Введите запрос
ВОЙТИ
Главная
В тренде
Подписки
Библиотека
История
Toyota RAV4 style
Реклама Toyota Russia
УЗНАТЬ БОЛЬШЕ
18:35
Построили ДОМ ПОД ЗЕМЛЕЙ своими РУКАМИ в ЛЕСУ —...
ЛЕСНЫЕ
1,8 млн просмотров
2 недели назад
ЗВЕЗДЫ МОЛОДЯТСЯ
COMEDY
17:27
Камеди Клаб Новый сезон Павел Воля «Звёзды...
Comedy Club
3 млн просмотров
1 день назад
4:49
вот такие добрые русские медведи на колыме
Алексей Трикачев
5,8 млн просмотров
7 лет назад
2:42
DANCE MONKEY METRO STATION PIANO...
Peter Buka
62 млн просмотров
7 месяцев назад
4:03
Выдры Котаро и Хана наслаждаются...
KOTSUMET
17 млн просмотров
1 год назад
8:17
Китайцы построили небесную дорогу
Техно
1,5 млн просмотров
1 месяц назад

А еще?


пульс
Профиль
Ваша персональная лента
Используя Пульс, вы принимаете Условия Использования
Кадетское братство: кто стоит за одним из крупнейших подрядчиков Москвы времен Собянина
Уже несколько лет в России действует таинственная компания с названием из трех букв, которой один за другим достаются огромные госконтракты. При этом владельцы
FORBES.RU
3

Что можно рекомендовать?

- Видео
- Музыку
- Статьи
- Книги
- Посты в социальных сетях

Теория

U - множество субъектов(users/пользователей/клиентов)
I - множество объектов(items/предметов/товаров/видео/треков)
Y - пространство описания транзакций
D - $(u_t, i_t, y_t)_{t=1}^{m} \in U \times I \times Y$ - транзакционные данные

Агрегированные данные:

$$R = ||aggr\{(u_t, i_t, y_t) \in D | u_t = u, i_t = i\}||$$

Задачи:

- прогнозирование незаполненных ячеек матрицы R
- оценивание сходства
- формирование списка рекомендаций для u или i

Рекомендательные системы на основе рейтинга

U - пользователи сайта

I - фильмы

r_{ui} = рейтинг, который пользователь u поставил фильму i

Задачи персонализации предложений:

- для пользователя:
  - выдать оценку фильму i
  - выдать ранжированный список рекомендованных фильмов
- для фильма:
  - выдать список фильмов, близких к i

Netflix Prize

1. 2006-2009
2. 100 млн оценок({1,2,3,4,5})
3. Задача - улучшить качество предсказания оценки пользователя на 10%
4. Приз - 1 000 000 $
5. метрика RMSE

Netflix Prize

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40

Выводы:

1) Можно делать композиции алгоритмов
2) Методов решения задачи много

- Конкурсы опасны - самые точные методы были слишком сложны для масштабирования и внедрения
- Метрика не очень:(

Специфика задачи построения рекомендательных систем

- Отсутствует признаковое описание
- Имеется в наличии очень малое количество данных
- Данные смещены в положительную сторону

Типы рекомендательных систем

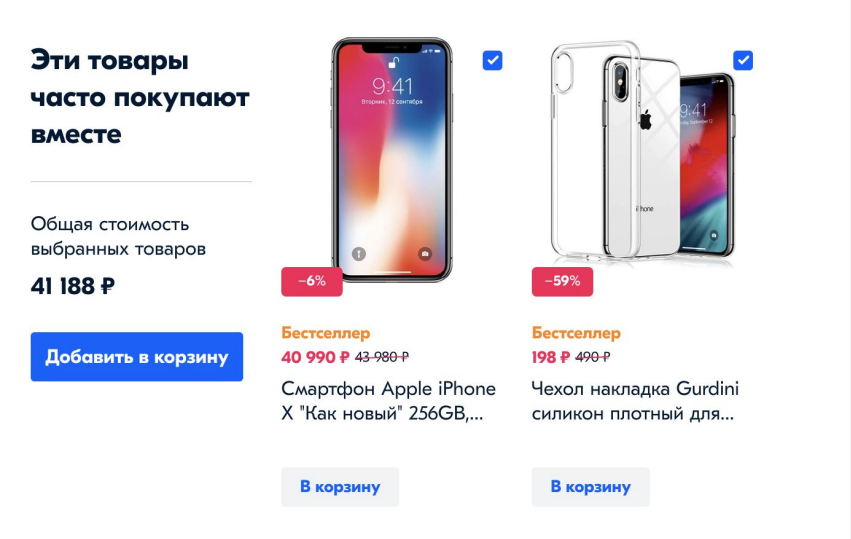
- Collaborative Filtering
- Content-Based systems
- Knowledge-based systems

Коллаборативная фильтрация

- Корреляционные модели
  - хранение всей исходной матрицы R
  - сходство пользователей - корреляция строк R
  - сходство предметов - корреляция столбцов R
- Model Based подход
  - оценивание скрытых характеристик(профилей) пользователей и предметов
  - хранение профилей вместо всей матрицы
  - сходство пользователей и предметов - сходство их профилей

Простая коллаборативная фильтрация

“Пользователи, покупавшие яблоки, также покупали туалетную бумагу”



Проблемы:

- Тривиальные рекомендации
- Не учитываются интересы пользователей
- Проблема холодного старта
- Надо хранить всю матрицу R

User Based

“Пользователи, похожие на этого пользователя, часто покупают яблоки”

$U(u_0) = \{sim(u_0, u) > \alpha\}$-коллаборация

$I(u_0) = \{i \in I | B(i) = \frac{U(u_0) \cap U(i)}{U(u_0) \cup U(i)} > 0\}$- множество кандидатов

Сортируем по B, берем топ - готово!

Проблемы User Based

- Проблема холодного старта
- надо хранить всю матрицу R
- нечего рекомендовать новым пользователям

Item Based

“Вместе с товарами, которые покупал Вася, часто покупают X”

$$X = \{i \in I | \exists i_0 : i_0 \in I(u_0) \text{ и } B(i) = sim(i, i_0) > \alpha\}$$

Сортируем по B, берем топ.

Проблемы Item Based

- тривиальность рекомендаций
- Проблема холодного старта
- надо хранить всю матрицу R

Меры похожести

- Корреляция Пирсона
- Косинусная мера
- статистические критерии:
  - χ^2 тест
  - тест Фишера
- Что-либо еще(специфика задачи)

Пример: Item2Item

Youtube(2010):

- Идем от похожести роликов
- Рассматриваем взаимодействия за некоторый период времени(сутки)
- c_{ij} - количество раз, когда лайкали/смотрели/долго смотрели ролик i с роликом j

$$r(i,j) = \frac{c_{ij}}{c_i c_j}$$

Корреляционные модели: резюме

Преимущества:

- Легко понять
- Легко реализовать

Недостатки:

- Не хватает теоретического обоснования(все вокруг эвристик)
- Требуется хранить большие объемы данных
- Проблема холодного старта

Латентные модели

Латентная модель: по данным мы пытаемся оценить:

$(p_{tu})_{t \in G}, |G| \ll |I|$ - профили пользователя

$(q_{ti})_{t \in H}, |H| \ll |U|$ - профили предметов

Типы моделей

- Ко-кластеризация:
  - жесткая p_{tu}, q_{ti} - 1 если принадлежит к кластеру, 0 - если нет
  - мягкая p_{tu}, q_{ti} - степени принадлежности к кластерам(EM алгоритм)
- Матричные разложения
- Вероятностные модели
- Нейронные сети

Матричные разложения

Есть множество интересов - T, профили представимы в виде матриц:

$$P = (p_{tu})_{|T| \times |U|}, Q = (q_{ti})_{|T| \times |I|}$$

Задача: найти разложение $r_{ui} = \sum_{t \in T} \pi_t p_{tu} q_{ti}$

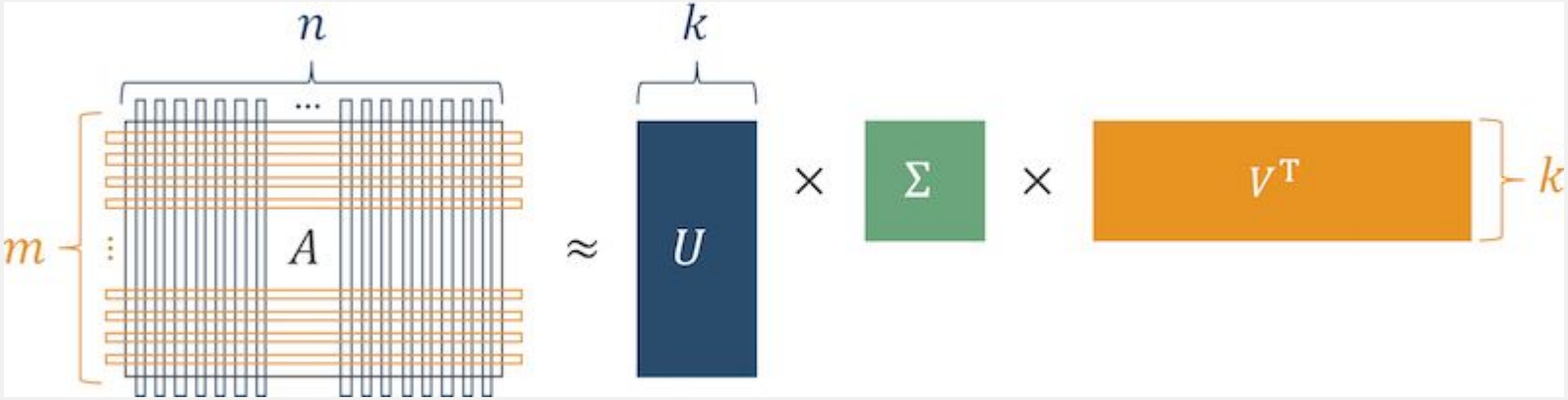
Или в матричном виде: $R = P^T \Delta Q, \Delta = diag(\pi_i)_{i \in T}$

Методы решения:

SVD - сингулярное разложение

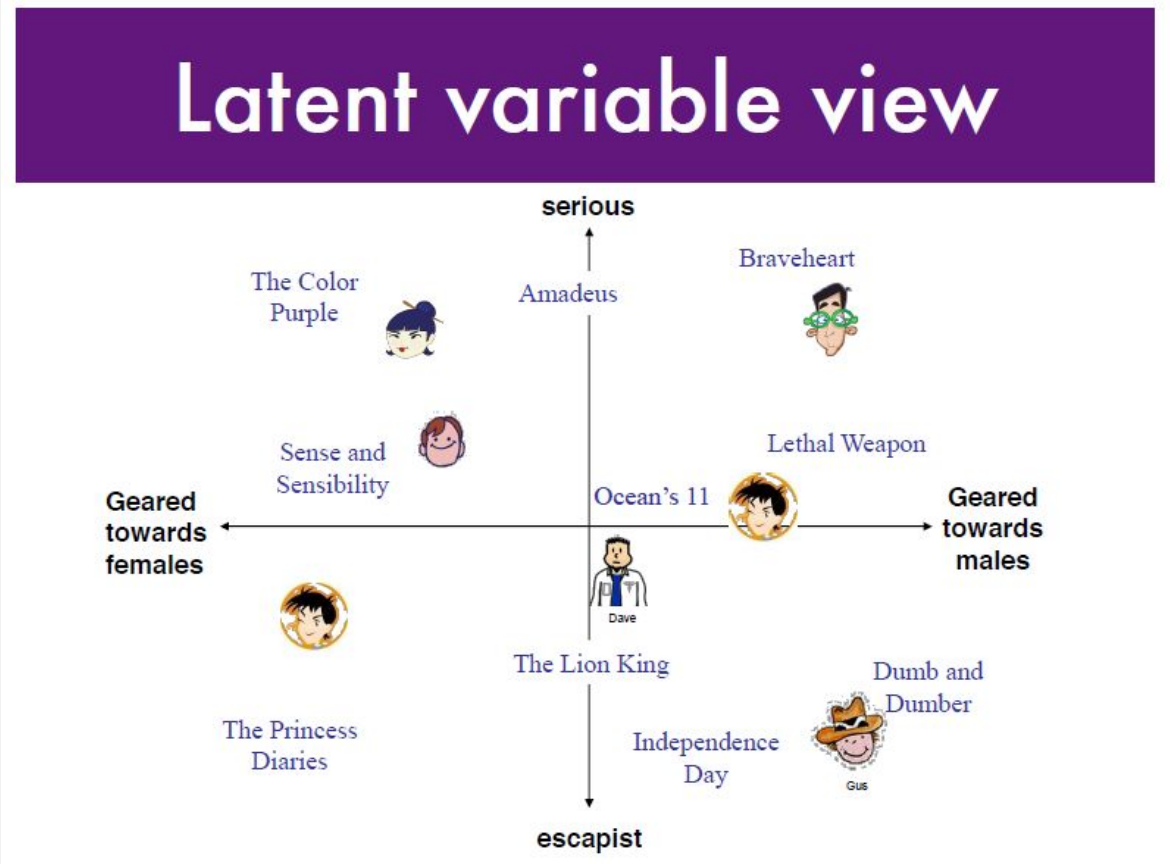
NNMF - неотрицательное матричное разложение

SVD разложение



Пытаемся разложить нашу матрицу на необходимые для нас профили

Латентные факторы



Как решаем

Постановка задачи: $||R - P^T Q||^2 \to \min_{P,Q}$

Используем SGD:

Пусть $\epsilon_{ui} = r_{ui} - p_u^T q_i$

Тогда:

$$p_u = p_u + \eta \epsilon_{ui} q_i$$
$$q_i = q_i + \eta \epsilon_{ui} p_u$$

Легко ввести регуляризацию: $+\lambda||P||^2 + \mu||Q||^2$

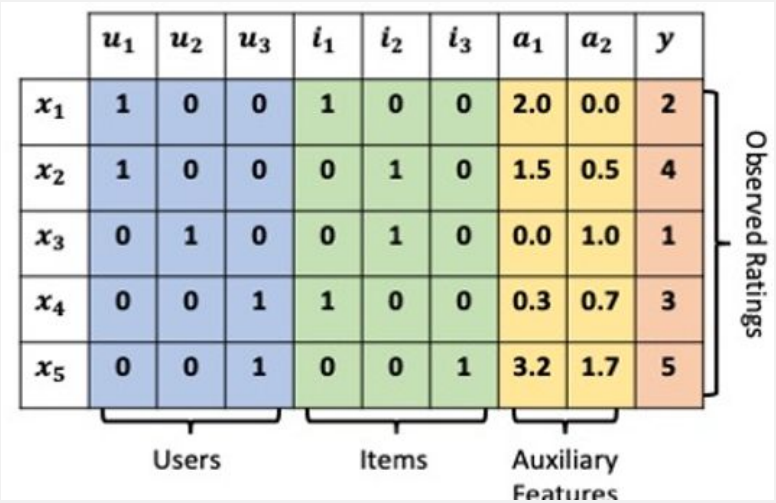
Отсутствие интерпретируемости

Линейные модели

Нет признаков - не беда, сделаем!

Признаки:

- Номер пользователя u (one-hot encoding)
- Номер объекта i (one-hot encoding)
- Любая дополнительная информация о объекте/пользователе(кол-во взаимодействий пользователя с другими объектами, кол-во взаимодействий других пользователей с этим объектом)



Factorization Machines

Факторизационные машины представлены как универсальная модель коллаборативной фильтрации, обобщающая многие из известных моделей:

$$h(x) = w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} V_j^T V_{j'}$$

$x \in R^p$ - вектор признаков объекта

$h(x)$- предсказание

Модель “квадратичной” регрессии

Factorization Machines

$$h(x) = w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} V_j^T V_{j'}$$

	u_1	u_2	u_3	i_1	i_2	i_3	a_1	a_2	y
x_1	1	0	0	1	0	0	2.0	0.0	2
x_2	1	0	0	0	1	0	1.5	0.5	4
x_3	0	1	0	0	1	0	0.0	1.0	1
x_4	0	0	1	1	0	0	0.3	0.7	3
x_5	0	0	1	0	0	1	3.2	1.7	5

Observed Ratings

Users | Items | Auxiliary Features

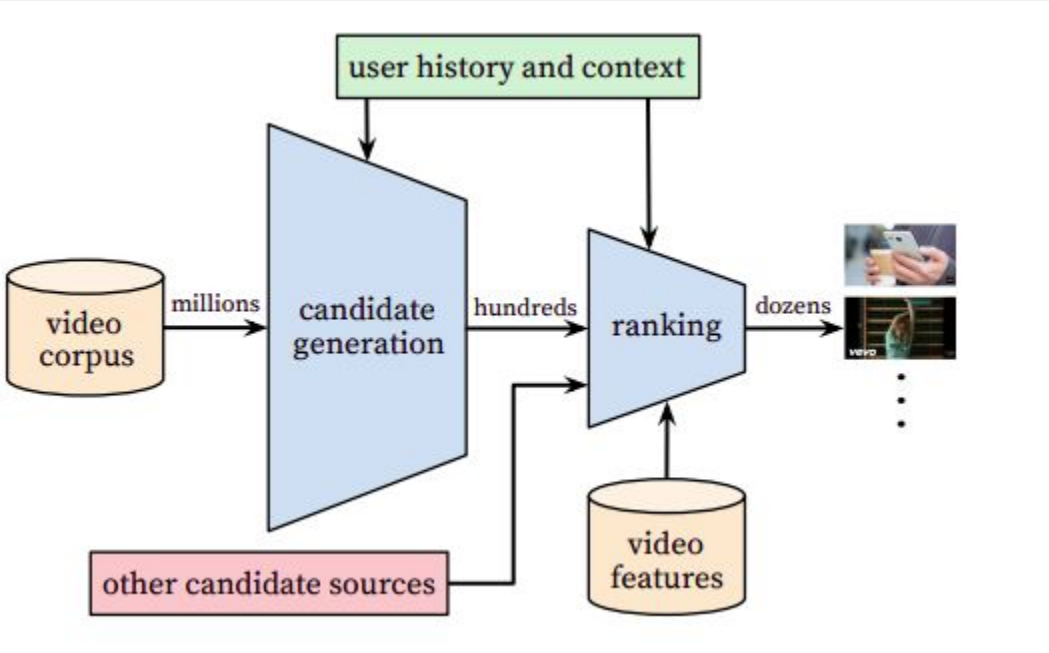
Factorization Machines

- Позволяет моделировать популярные модели коллаборативной фильтрации
- Позволяет добавить новые признаки(контекст)
- Имеет эффективный алгоритм обучения
- Куча opensource библиотек

Model-Based

- Обучение моделей в оффлайне
- Модели надо часто переобучать и обновлять
- Проблема подбора кандидатов

Процесс рекомендаций



Генерация кандидатов

- Топ по кликам/покупкам
- Топ по интересу пользователя
- Похожие на просмотренные
- Approximate Nearest Neighbour Search
- Заготовленные заранее самые “важные” объекты для пользователя

Метрики

- Математические:
  - MAE
  - RMSE
  - Precision@K
  - Recall@K
  - MAP@K
  - DCG@K
- Качественные метрики:
  - Ручная разметка(side-by-side, user)

Метрики

- MAE

$$MAE = \frac{1}{k}\sum_{i=1}^{k}|r_i - y_i|$$

- RMSE

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(r_i - y_i)^2}$$

Метрики

DCG

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

А еще важно:

- Разнообразие
- Неожиданность
- Свежесть
- Удобство в использовании
- Доверие
- Кликбейт, желтизна

Открытые вопросы

- Как обосновывать рекомендации?
- Как решать проблему холодного старта?
- Как учитывать контекст?
- Как учитывать неявные предпочтения?
- Как помогать выбираться из “пузыря”?
- Как учитывать связи между пользователями?

Технические вопросы

- Как быстро перестраивать рекомендации?
- Как масштабировать?
- Как отбирать кандидатов?

ТЕХНОСФЕРА

Спасибо за внимание!

Вопросы?