

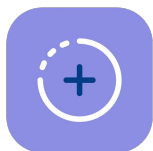


# Контентные методы рекомендаций.



Проверить, идет ли запись

# Меня хорошо видно && слышно?



Ставим “+”, если все хорошо  
“-”, если есть проблемы



# Маршрут вебинара



Контентные подходы

FM

Ранжирующие функции потерь

LightFM

Практика





# Цели вебинара

К концу занятия вы

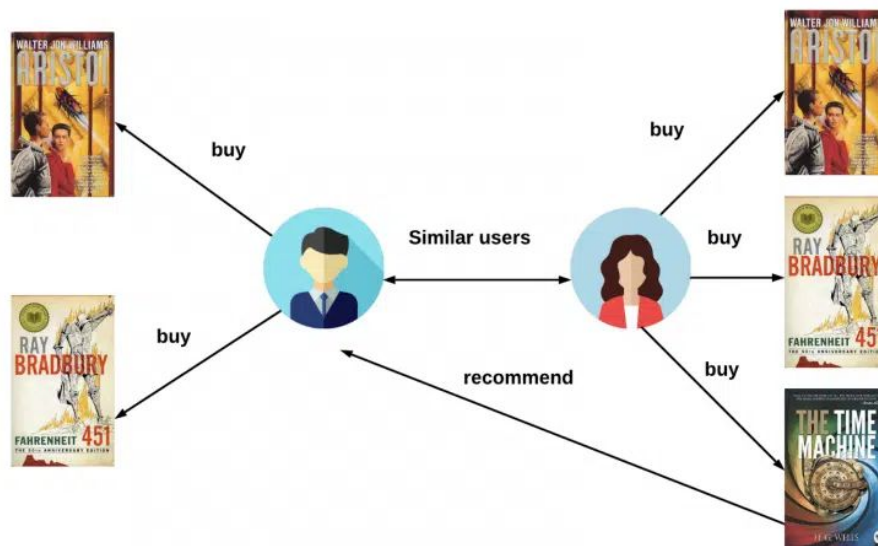
1. Изучить особенности контентных методов рекомендаций
2. Познакомиться с ранжирующими функциями потерь
3. Научиться применять MF и LightFM - модели

# Доступная информация о товарах

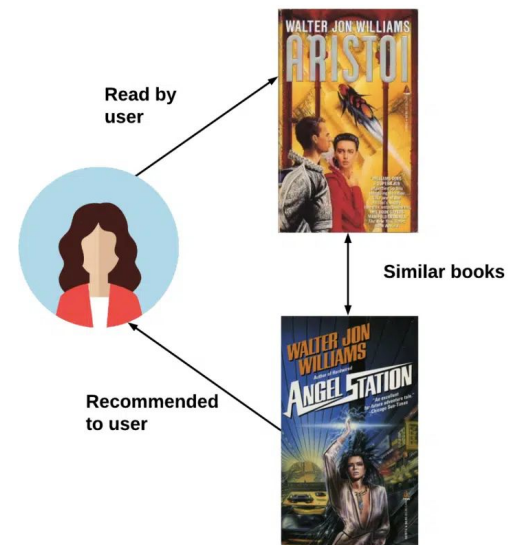
- взаимодействия
- признаки пользователей и объектов: числовые, категориальные характеристики, теги, изображения, текст и т. д.
- контекст: дата и время, устройство, погода и т. д.

 <p><b>44 Р</b> 55 Р -20%</p> <p>Вода негазированная Святой Источник, 1,5 л</p>	 <p><b>255 Р</b> 306 Р -17%</p> <p>Витамин Е жидкий Mirrolla, 50 мл</p>
 <p><b>654 Р</b> 1501 Р -56%</p> <p>CUP7 Жидкое Средство для очистки капучинатора и м...</p>	 <p><b>1112 Р</b> 2812 Р -60%</p> <p>Осталось 901 шт</p> <p>Кухонный смеситель, предназначенный для кухо...</p>

# Коллаборативная vs контентная фильтрация

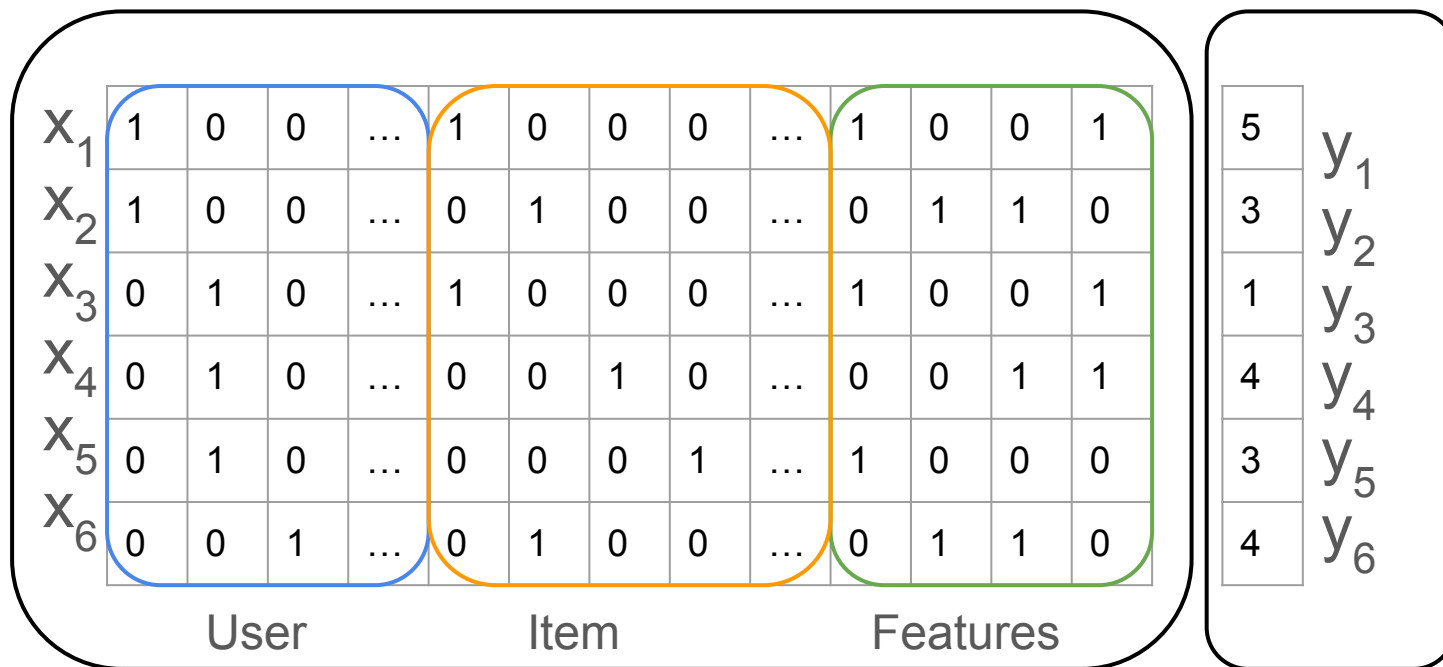


- Не нужны признаки объектов и пользователей
- Разнообразие рекомендованных объектов
- Проблема холодного старта



- Нужны признаки объектов
- Рекомендации могут быть однообразными
- Нет проблемы холодного старта для объектов

# Factorization Machines (FM)



User - индикатор пользователя.

Item - индикатор объекта.

Features - признаки.

Target - рейтинг.

# Factorization Machines (FM)

## [Статья](#)

FM учитывает признаки, а также их попарное взаимодействие.

$$\hat{y}(x) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j$$

$w_0$  - глобальное смещение

$w_i$  - веса каждого признака вектора  $x$

$\hat{w}_{i,j} = \langle v_i, v_j \rangle$  - скалярное произведение векторов,

учитывает взаимодействие между  $i$ -м и  $j$ -м признаком.



# Factorization Machines (FM)

## Преимущества

- использует признаки и их комбинации
- подходит для холодных объектов и пользователей
- можно использовать различные функции потерь

## Недостатки

- линейная зависимость для числовых признаков
- нет векторного представления пользователя и элемента и нет быстрого инференса
- редко используется в реальных рекомендательных системах

## Имплементации

[LibFM](#)

[RankFM](#)

# Функции потерь

- регрессия (например, среднеквадратическая ошибка)
- классификация (например, бинарная кросс-энтропия, кросс-энтропия)
- ранжирование (например, Triplet Loss, BPR, WARP)

Основная идея - переход от предсказания значения рейтинга к предсказанию правильного порядка ранжирования.

# Задача ранжирования

Не понравившиеся фильмы  
(с отрицательным ранком)

Еще не просмотренные

Понравившиеся фильмы  
(с положительным ранком)

Прометей



Китайский синдром



Меланхолия



...



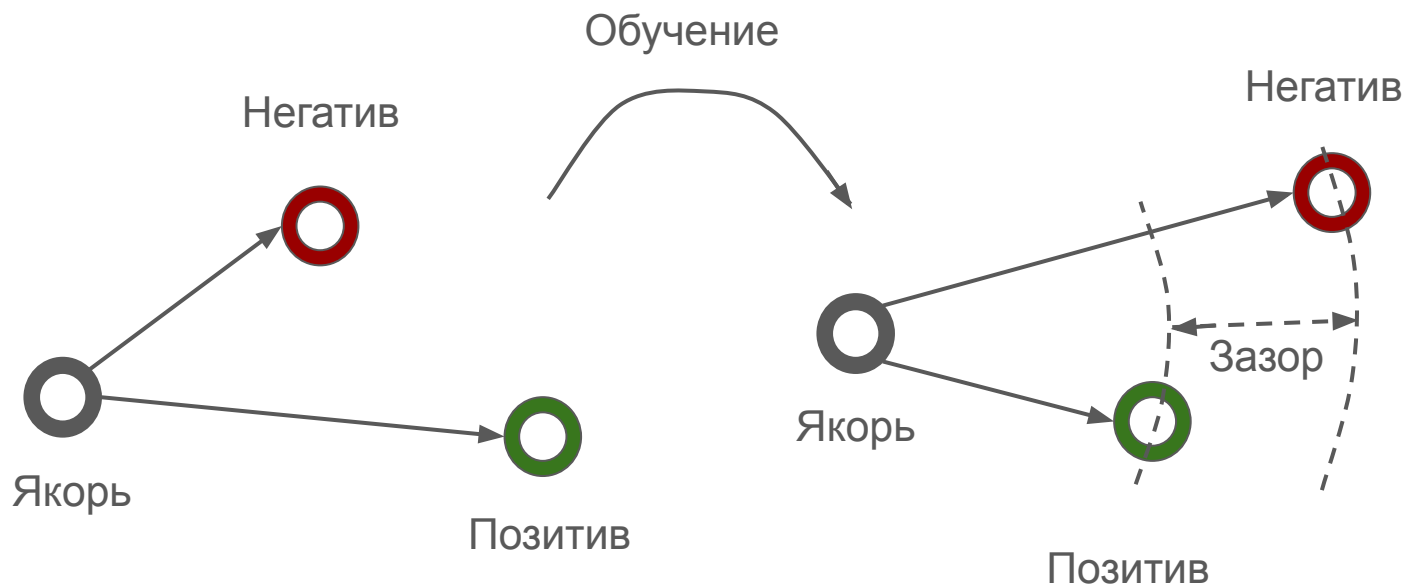
Притяжение



Интерстеллар



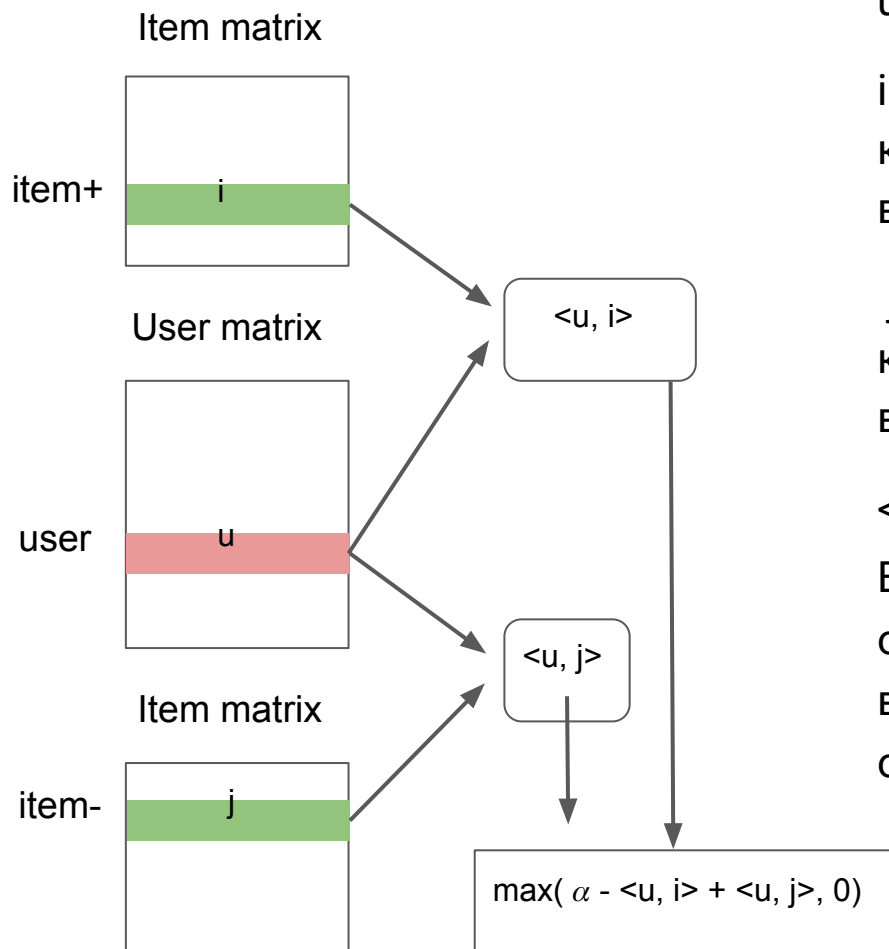
# Triplet Loss



Triplet Loss изначально начал применяться в задачах CV.

Он минимизирует расстояние между якорем и позитивом, оба из которых имеют одинаковую идентичность, и максимизирует расстояние между якорем и негативом.

# Triplet Loss в задачах рекомендации



$u$  - вектор пользователя

$i$  - вектор положительного объекта, с которым пользователь ранее взаимодействовал

$j$  - вектор отрицательного объекта, с которым пользователь ранее не взаимодействовал

$\langle , \rangle$  - скалярное произведение

Вектор пользователя и положительного объекта должны быть более похожи, чем вектор пользователя и вектор отрицательного объекта.

# Bayesian Personalized Ranking (BPR)

## [Статья](#)

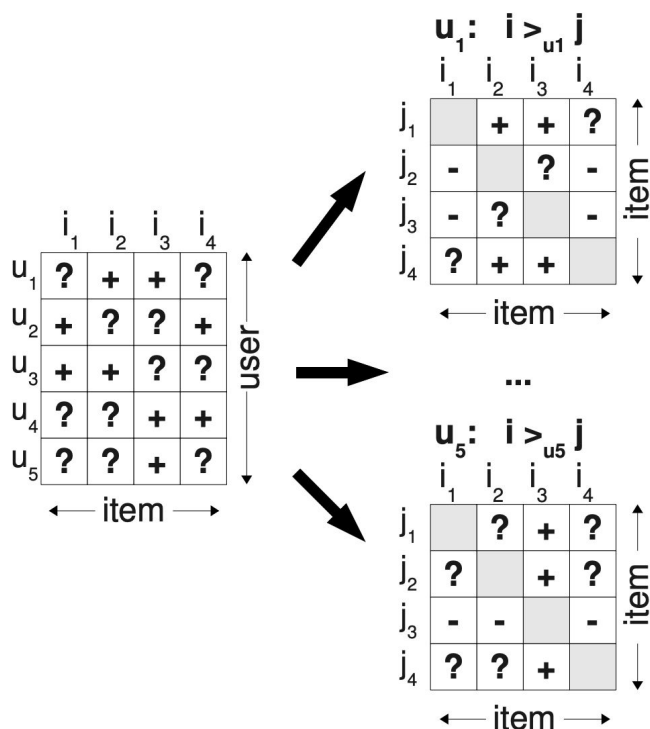
Формализация задачи:

$U$  - множество пользователей;

$I$  - множество объектов

Все пары  $(u, i)$  - это известные взаимодействия  $S$ . Задача ранжировать такие взаимодействия выше, чем отсутствующие в данных взаимодействия  $(U \times I) \setminus S$ .

# Bayesian Personalized Ranking (BPR)



Пользователь  $u_1$ :

- взаимодействовал с объектами  $i_2$  и  $i_3$ ,
- не взаимодействовал объектами с  $i_1$  и  $i_4$ .

Объекты, с которыми пользователь взаимодействовал, предпочтительнее тех, с которыми он не взаимодействовал.

# Bayesian Personalized Ranking (BPR)

Алгоритм обучения:

1. Выбираем пользователя  $u$
2. Берем для него объект  $i$ , с которым он взаимодействовал
3. Семплируем случайный объект  $j$ , с которым он не взаимодействовал
3. Вычисляем разность скоров для этих объектов  $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$
4. Делаем обновление весов модели



# Weighted Approximate-Rank Pairwise (WARP)

[Статья](#)

Выход модели для рассматриваемого элемента минус порог



# Weighted Approximate-Rank Pairwise (WARP)

Алгоритм обучения:

1. Выбираем пользователя  $u$
2. Берем для него объект  $i$ , с которым он взаимодействовал
3. Семплируем случайный объект  $j$ , с которым он не взаимодействовал
4. Вычисляем разность скоров для этих объектов
5. - Если отрицательный объект нарушает порядок ранжирования с заданным порогом, то делаем обновление весов модели
  - Если не нарушает, то переходим к шагу 3
  - Если не удастся найти объекты, нарушающие порядок ранжирования, то переходим к шагу 2, а затем к шагу 1

# LightFM

## [Статья](#)

LightFM объединяет коллаборативную фильтрацию с контентными подходами.

Формализация задачи:

$U$  - множество пользователей;

$I$  - множество объектов

$F^U$  - множество признаков пользователей;

$F^I$  - множество признаков объектов.

Все пары  $(u, i)$  - это объединение положительных  $S^+$  и отрицательных взаимодействий  $S^-$ . Отрицательные примеры можно сэмплировать из множества объектов, с которыми пользователь не взаимодействовал.

Каждый пользователь имеет набор признаков  $f_u \subset F^U$ , а объект -  $f_i \subset F^I$ .

# LightFM

Для каждого признака  $f$  задается вектор размерности  $d$  отдельно для пользователей и объектов.

Латентное представление пользователя  $q_u = \sum_{j \in f_u} e_j^U$

Латентное представление объекта  $p_i = \sum_{j \in f_i} e_j^I$

Смещения (bias) обучаются отдельно  $b_u = \sum_{j \in f_u} b_j^U, b_i = \sum_{j \in f_i} b_j^I$

Предсказания получаются в виде скалярного произведения

с поправкой на смещения пользователя и объекта  $\hat{r}_{ui} = f(q_u \cdot p_i + b_u + b_i)$

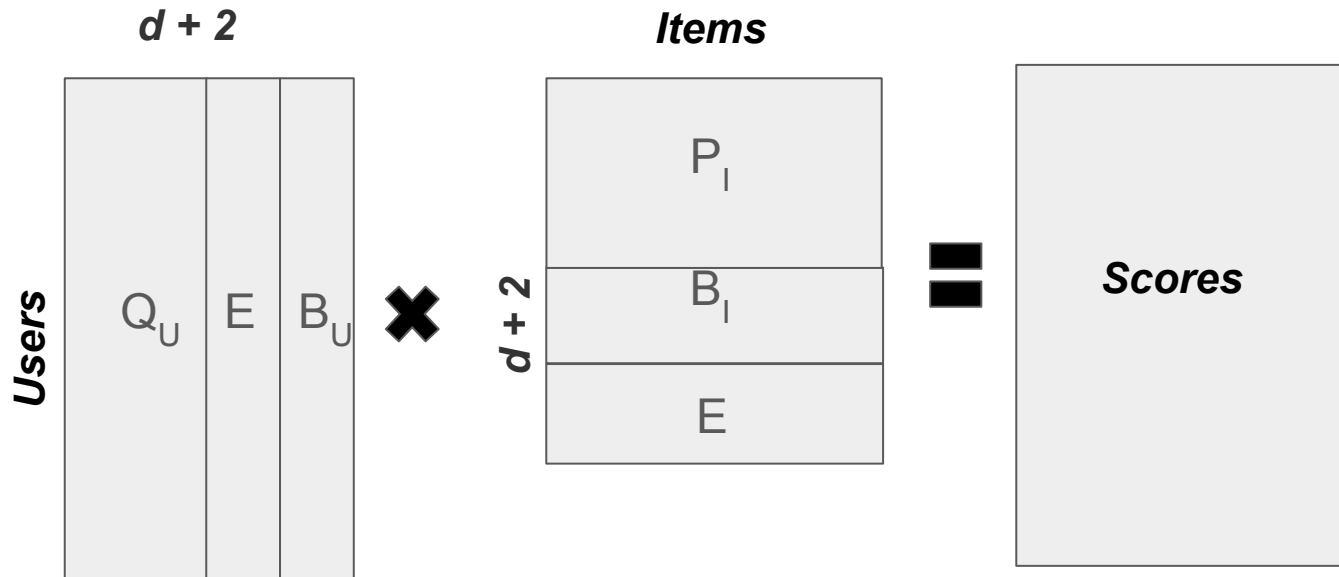
# LightFM

Эмбединги пользователей и эмбединги объектов можно обучать как с использованием их признаков, так и без.

Если признаки используются, то эмбединги пользователей и объектов представляют собой сумму векторов их признаков (включая id как признак).

Холодные пользователи и объекты могут быть представлены как суммы векторов их признаков.

# LightFM



В векторном виде можно получить предсказания следующим образом:  
конкатенировать матрицу эмбеддингов с единичным вектором и вектором  
смещения.

# LightFM

## Преимущества

- обучаются векторы пользователя и объекта для быстрого инференса
- популярная библиотека с несколькими функциями потерь
- может обучаться как с признаками, так и без
- подходит для холодных объектов и пользователей
- при обучении без признаков показывает качество сравнимое с другими моделями матричной факторизации

## Недостатки

- вектор пользователя представляет собой линейную комбинацию его признаков

## Имплементация

[LightFM](#)

# Вопросы?



Ставим “+”,  
если вопросы есть



Ставим “-”,  
если вопросов нет



# Рефлексия

# Цели вебинара

## Проверка достижения целей

1. Изучить особенности контентных методов рекомендаций
2. Познакомиться с ранжирующими функциями потерь
3. Научиться применять MF и LightFM - модели