

# Recsys Intro

Малышев Сергей  
Одноклассники

## Plan

- Что такое рек. системы и зачем они нужны
- Постановки задач рекомендаций
- Типы рекомендательных систем
  - Неперсонализированные
  - Контентные
  - Коллаборативная фильтрация
  - Гибридные модели
- Метрики качества рекомендаций

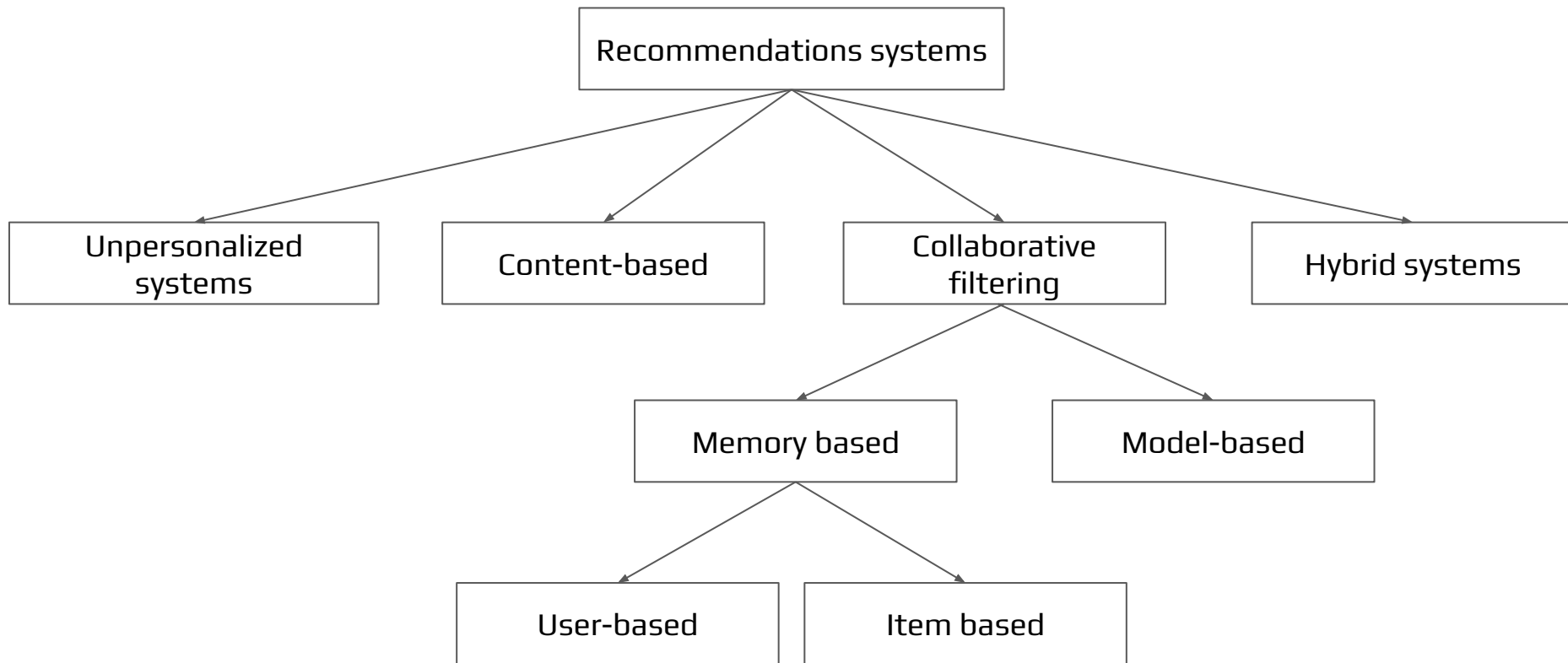
## Что такое рек. системы и зачем они нужны

- Системы, которые могут предлагать пользователям релевантный контент, который он сам бы не смог найти из-за того, что его очень много.

## Постановки задач рекомендаций

- Item2user
- Item2item
- user2item

# Типы рекомендательных систем



## Неперсонализированные рекомендации

- Рекомендации одинаковые для всех пользователей
  - По популярности
  - Новизне
  - Цене
  - CTR-у
  - По отзывам/рейтингам
  - Подборки

## Неперсонализированные рекомендации

- Pros:

- Очень легко завести и это дает предсказуемый и стабильный результат
- Решает проблему холодного старта
- Разбавление рекомендаций

- Cons:

- Не персонализированные, низкая релевантность по сравнению с персонализированными

## Контентные рекомендации

- В отличие от неперс. рекомендаций рекомендации уже строятся разные на основе контента.
- Бывают
  - Персональными, то есть они учитывают историю пользователя
  - Неперсонализированными, то есть строятся только на контентной информации (например item2item только на контентных признаках)



## Контентные рекомендации

Как будем рекомендовать:

- Используем свойства айтема (признаки из текста, категорию, и тд) и свойства пользователя (например соц дем и гео) сконкатинируем признаки в одну матрицу признаков
- Добавим лейбл релевантности контента
- Решаем задачу классификации
- Так же можно рекомендовать контент близкий по расстоянию к тому контенту, с которым пользователь уже взаимодействовал

## Контентные рекомендации

- Proc
  - Просто
  - Интерпретируемо
  - Решает проблему холодного старта
- Cons
  - Держит в контентном пузыре

## Коллаборативная фильтрация

### Мотивация:

- Хотим делать еще более персонализированные рекомендации без привязки к свойствам айтемов и юзеров.
- Хотелось бы учесть некоторую “похожесть” пользователей.



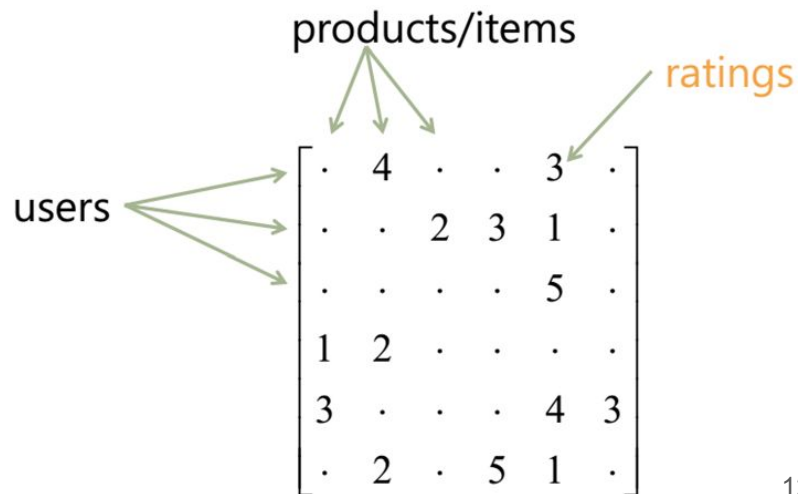
CF: система рекомендаций, построенная на фидбеках пользователей без знания свойств пользователей и объектов.

# Коллаборативная фильтрация

## Постановка задачи:

- Имеем матрицу user-item со значениями релевантности айтема данному пользователю, в которой есть пропущенные значения
- Задача восстановить пропущенные значения в матрице

$$F: User \times Item = Utility (Relevance)$$



# Коллаборативная фильтрация

## Explicit, implicit feedback

### Explicit

+: Явное действие, коррелирует с его предпочтениями

- : Мало
- : Нужны дополнительные действия от пользователя
- : Не везде он есть
- : Он только в контексте (в данный момент времени)

### Implicit

+: Есть везде и его много

- : Нет точной уверенности, что это положительный сигнал и отражает предпочтения пользователя

## Коллаборативная фильтрация

- Memory-based: запоминаем некоторую историческую информацию и сразу же делаем рекомендации.
  - User-based
  - Item-based
- Model-based: на исторических данных обучаем модель, которая будет рекомендовать пользователям самые релевантные айтемы.
  - Matrix-Factorization
  - Weighted-regularized matrix factorization (ALS)

## Memory based (User-based)

Ищем похожих на меня пользователей и  
рекомендуем мне то же самое что и им.

Базовый подход:

$$r_{ui} = \frac{\sum_{v \in U} sim_{uv} \cdot r_{vi}}{\sum_{v \in U} sim_{uv}}, \quad U - \text{мн} - \text{во пользователей}$$

(не учитывает разные шкалы оценок пользователей)

Нормализованный:

$$r_{ui} = \overline{r_u} + \sigma_u \frac{\sum_{v \in U} sim_{uv} \cdot \frac{r_{vi} - \overline{r_v}}{\sigma_v}}{\sum_{v \in U} sim_{uv}}, \quad U - \text{мн} - \text{во пользователей}$$

## Memory based (Item-based)

Все то же самое, только теперь ищем не похожих пользователей, а похожие айтемы

$$r_{ui} = f^{-1} \left( \frac{\sum_{j \in I} sim_{ji} \cdot f(r_{uj})}{\sum_{j \in I} sim_{ji}} \right), \text{ } I - \text{мн. во айтемов}$$



Memory based

Как искать похожесть?

Можно использовать косинусное расстояние

$$sim_{uv} = \frac{\sum_{i \in I} \left( \frac{r_{ui} - \bar{r}_u}{\sigma_u} \right) \left( \frac{r_{vi} - \bar{r}_v}{\sigma_v} \right)}{\sqrt{\sum_{i \in I} \left( \frac{r_{ui} - \bar{r}_u}{\sigma_u} \right)^2} \sqrt{\sum_{i \in I} \left( \frac{r_{vi} - \bar{r}_v}{\sigma_v} \right)^2}}, \quad I - \text{мн-во айтемов}$$

# Memory based

## User-based vs Item-based

	User-based	Item-based
Точность (релевантность)	Точнее, если пользователей гораздо меньше чем айтеров	Точнее, если айтеров гораздо меньше чем пользователей
Стабильность	Стабильнее, если изменения в количестве пользователей не значительны	Стабильнее, если изменения в количестве айтеров не значительны
Интерпретируемость	-	+
Разнообразие	+	-

## Memory based

### Proc:

- Простота и интерпретируемость
- Легковесность и быстрота

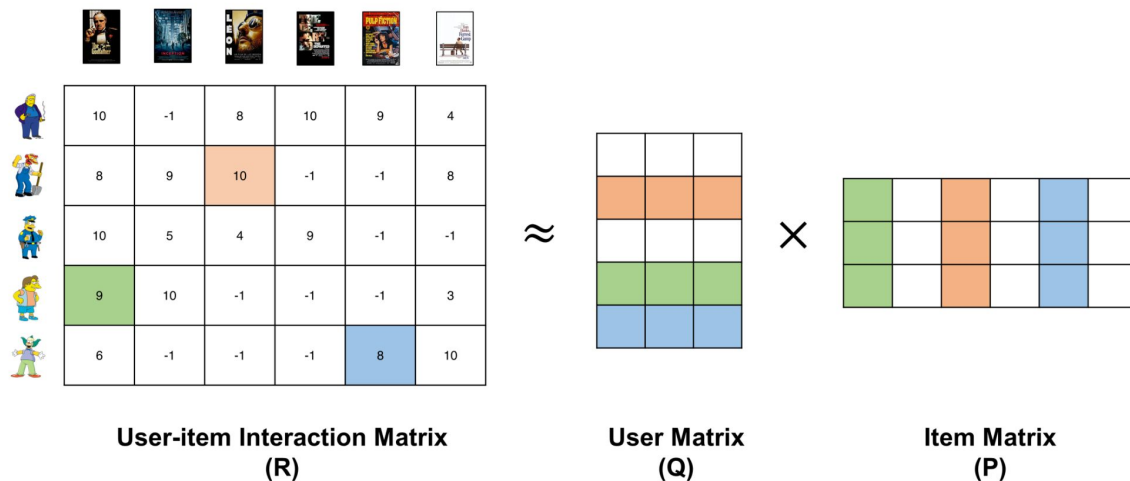
### Cons:

- Частичное покрытие каталога
- Cold-start
- Не может найти сложные зависимости

## Model based подход

Раскладываем user-item матрицу на матрицу пользователей размера (n x k) и матрицу айтеров размера (k x m), где n - количество пользователей, m - айтеров, k - размерность латентных признаков

$$r_{ui} = x_u^T \cdot y_i$$



## Model based: matrix factorization

Постановка задачи:

$$L = \sum (r_{ui} - x_u^T \cdot y_i)^2 + \lambda \left( \sum \|x_u\|^2 + \sum \|y_i\|^2 \right) \rightarrow \min$$



Разложение



Регуляризация

## Model based: matrix factorization

### Алгоритм:

- Заполняем матрицы X и Y случайными значениями
- Выбираем случайную пару пользователя u из X и айтема i из Y
- Считаем градиенты:

$$\begin{cases} x_u \leftarrow x_u + \eta \cdot (e_{ui} \cdot y_i - \lambda \cdot x_u) \\ y_i \leftarrow y_i + \eta \cdot (e_{ui} \cdot x_u - \lambda \cdot y_i) \end{cases}, e_{ui} = r_{ui} - \hat{r}_{ui}$$

- Несколько итераций -> profit

## Model based: ALS

Постановка задачи:

$$L = \sum c_{ui} (p_{ui} - x_u^T \cdot y_i)^2 + \lambda \left( \sum \|x_u\|^2 + \sum \|y_i\|^2 \right) \rightarrow \min$$

$$p_{ui} = \begin{cases} 1, & r_{ui} > 0 \\ 0, & otherwise \end{cases} \quad - \text{индикатор совершенного действия}$$

$$c_{ui} = \begin{cases} 1 + \alpha \cdot r_{ui}, & r_{ui} > 0 \\ 1, & otherwise \end{cases} \quad - \text{уверенность в данном действии}$$

## Model based: ALS

Интуиция: мы все также хотим раскладывать матрицу на матрицу пользователей и айтемов и моделировать уже индикатор совершенного неявного действия минимизируя ошибку между историческим индикатором совершенного действия и предсказанным. При этом мы понимаем, что если действие не было совершено, это не значит, что оно не произойдет => наш confidence в этом минимальный. Таким образом мы хотим чтобы модель больше обращала внимания на события с высокой уверенностью, соответственно штраф на этих парах будет выше.



## Model based: ALS

### Оптимизация:

- Так как у нас есть все элементы матрицы, мы можем на один шаг положить одну из матриц константной и решать оптимизационную задачу только с одной из матриц

$$\begin{cases} x_u = (Y^T \cdot C^u \cdot Y + \lambda \cdot I)^{-1} \cdot Y^T \cdot C^u \cdot p(u) \\ y_i = (X^T \cdot C^i \cdot X + \lambda \cdot I)^{-1} \cdot X^T \cdot C^i \cdot p(i) \end{cases},$$

$C^u, C^i$  – диагональные матрицы размера  $u \times u, i \times i, C_{ii}^u = c_{ui}, C_{ui}^i = c_{ui}$

- Попеременно считая  $X$  и  $Y$  за константу, находим аналитическим образом оптимальные значения для  $X$  и  $Y$

## Model based

### Proc:

- Промышленный стандарт
- Точнее, чем memory-based

### Cons:

- Сложно интерпретировать
- Cold-start
- Линейная модель
- Модель надо обучать

## Метрики качества рекомендаций

$$precision @k = \frac{1}{k} \sum_{i=1}^k rel_i, rel_i - \text{релевантность на данном айтеме}$$

$$recall @k = \frac{1}{N} \sum_{i=1}^k rel_i, rel_i - \text{item relevance, } N - \text{total relevances}$$

$$AP @k = \frac{1}{k} \sum_{i=1}^k precision @i \cdot rel_i, rel_i - \text{item relevance}$$

$$MAP @k = \frac{1}{U} \sum_{i=1}^U AP_i @k, U - \text{total number of users}$$

## Метрики качества рекомендаций

$$DCG @k = \sum_{i=1}^k Gain @i \cdot penalty @i$$

$$Gain @i = 2^{rel @i} - 1$$

$$penalty @i = \frac{1}{\log_2(i + 1)}$$

$$nDCG @k = \frac{DCG @k}{\max(\{DCG @i\}_{i=1}^k)}$$

$$MRR @k = \frac{1}{U} \sum_{i=1}^U RR @k$$

$$RR @k = \frac{1}{\text{minimal relevant position}}$$

$$ROC - AUC$$

## Summary

Покрыли следующий пласт базовых подходов:

Рек. системы:

- Неперсонализированные
- Контентные
- User-based, item-based CF
- MF, ALS
- Метрики

Спасибо за внимание!