

Введение в рекомендательные СИСТЕМЫ

RecSys

lecturer: Mollaev D. E.
Sber AI Lab

Информация о курсе

- 6 лекций и семинаров
- 2 домашние работы:
 - HW1: практика
 - HW2: теория
- Формула оценивания:
 - $0.5 * HW1 + 0.5 * HW2$
- Дедлайны жесткие
- Экзамена нет

Github курса: https://github.com/Dzhambo/RecSys_MDS

Почта для вопросов и сдачи домашних работ: dmollaev@hse.ru

Структура курса

- Введение в рекомендательные системы
- Методы на основе матричных факторизаций, часть 1
- Методы на основе матричных факторизаций, часть 2
- Нейросетевые и content-based рекомендации, часть 1 (- HW1)
- Нейросетевые и content-based рекомендации, часть 2 (- HW2)
- Оценка качества рекомендаций

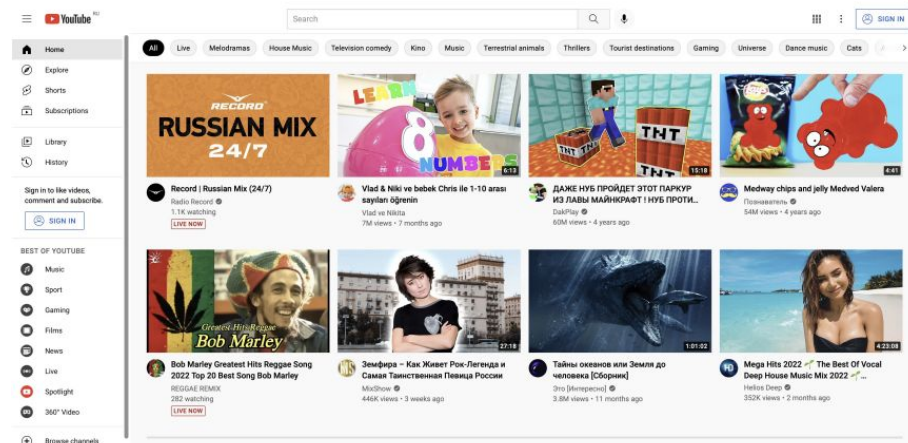
План лекции:

- Постановка задачи
- Коллаборативная фильтрация
- Контентные рекомендации
- Рекомендательные системы на практике
- Особенности рекомендательных систем

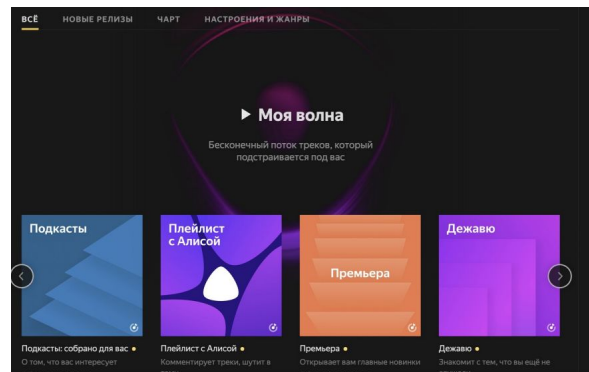
План лекции:

- Постановка задачи
- Коллаборативная фильтрация
- Контентные рекомендации
- Рекомендательные системы на практике
- Особенности рекомендательных систем

Зачем нужны рекомендательные системы?



Покупают вместе



Зачем нужны рекомендательные системы?

Пользователям:

- Более релевантные предложения
- Снижение негатива от лишних рекомендаций
- Экономия времени и сил на поиск
- Положительный клиентский опыт от использования сервиса

Компаниям:

- Вовлеченность пользователей
- Лояльность пользователей
- Улучшение клиентского опыта
- Дополнительная монетизация

Постановка задачи рекомендательных систем

Дано:

- U — множество пользователей.
- I — множество объектов, которые подлежат рекомендации (айтемов).

Для каждого пользователя $u \in U$ задана его история взаимодействий с множеством объектов $I_u \subset I$ в виде оценок:

$$R_u = (r_{ui})_{i \in I_u}$$

Такие оценки называются **фидбеком**.

Цель — предложить пользователю новые для него объекты, с которыми он ранее не взаимодействовал, но которые, с высокой вероятностью, будут ему **интересны**.

Постановка задачи рекомендательных систем

Как понять что item интересен пользователю?

- Товар положили в корзину
- Дослушали музыку до конца
- Лайкнули пост или статью
- Посмотрели хотя бы половину видео

Explicit feedback

Это действия, которые явно демонстрируют отношение пользователя к товарам и контенту, например:

- Оценка фильма по рейтингу.
- Лайк или дизлайк трека или видео.
- Написание рецензии на товар, статью или фильм.

Особенности:

- Данные такого типа встречаются редко.
- Дизлайк товара не всегда означает, что пользователю не стоит рекомендовать подобные товары — возможно, ему интересна сама категория, но конкретный товар не подошел.

Implicit feedback

Implicit feedback — это любые другие данные о действиях пользователя на сайте. Они не выражают его явного отношения к товару, но могут служить прокси для **explicit feedback**.
Например:

- Покупка товара
- Просмотр статьи
- Время просмотра видео и прослушивания музыки


























Особенности:

- Таких данных больше
- Доверия к такому фидбеку меньше
- Техники учитывающие такой фидбек имеют отличную природу от explicit feedback

План лекции:


















- Постановка задачи
- Коллаборативная фильтрация
- Контентные рекомендации
- Рекомендательные системы на практике
- Особенности рекомендательных систем

Коллаборативная фильтрация

Что стоит под знаком вопроса?

Коллаборативная фильтрация

Item \ User	1	2	3	4	5	6	7	8	9	10
Петя										
Маша										
Вася										
Катя										

- Что порекомендовать Кате?
- Что лучше не рекомендовать Пете?

Коллаборативная фильтрация

Идея:

- Мы стремимся рекомендовать пользователю товары, которые понравились другим похожим пользователям.
- Альтернативный подход — предлагать объекты, схожие с теми, которые уже понравились пользователю. Сходство между объектами можно определить на основе пользовательских взаимодействий.

В широком смысле коллаборативная фильтрация — это группа методов рекомендаций, которые используют сходство в истории взаимодействий между пользователями и товарами.

User2user рекомендации

1. Определение соседей

Для каждого пользователя u вводится мера схожести $s(u, v)$ с другими пользователями.

Соседями пользователя u считаются все пользователи v , для которых выполняется условие:

$$N(u) = \{v \in U \setminus \{u\} \mid s(u, v) > \alpha\}$$

2. Оценка предпочтений

Пользователю u рекомендуются объекты, которые получили высокие оценки у его соседей.

Для этого оцениваются рейтинги непросмотренных объектов на основе оценок соседей.

3. Формирование рекомендаций

После вычисления рейтингов для непросмотренных объектов выбирается несколько с наивысшими значениями, которые и рекомендуются пользователю.

Обозначения:

- I_u — множество объектов, которые пользователь u уже оценил.
- \bar{r}_u — средний рейтинг, выставленный пользователем u .

User2user рекомендации

Оценка рейтинга на основе соседей

Рейтинг объекта для пользователя можно оценить как средневзвешенное значение оценок его соседей:

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} s(u, v) r_{vi}}{\sum_{v \in N(u)} |s(u, v)|}$$

User2user рекомендации

Коррекция на индивидуальные шкалы пользователей

Разные пользователи могут использовать различные шкалы оценок. Чтобы учесть это, вводится поправка на средний рейтинг каждого пользователя:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} s(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |s(u, v)|}$$

User2user рекомендации

Нормализация рейтингов пользователей

Помимо различий в шкалах, оценки разных пользователей могут находиться в разных диапазонах. Чтобы это учесть, можно нормализовать их с учетом дисперсии:

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in N(u)} s(u, v)(r_{vi} - \bar{r}_v) / \sigma_v}{\sum_{v \in N(u)} |s(u, v)|}$$

Где σ_u — стандартное отклонение оценок пользователя u , вычисляемое по формуле:

$$\sigma_u = \sqrt{\frac{1}{|I_u|} \sum_{i \in I_u} (r_{ui} - \bar{r}_u)^2}$$

User2user рекомендации

Как выбрать $s(u, v)$?

1. Мера Жаккара (Jaccard Similarity)

Определяет степень пересечения множеств оцененных товаров у двух пользователей:

$$s(u, v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|}$$

Где:

- I_u — множество товаров, оцененных пользователем u .
- I_v — множество товаров, оцененных пользователем v .

User2user рекомендации

Как выбрать $s(u, v)$?

Вычисляется как сумма произведений соответствующих оценок пользователей для одних и тех же товаров:

$$s(u, v) = \sum_{i \in I_u \cap I_v} r_{ui} r_{vi}$$

Где r_{ui} — оценка пользователя u для товара i , а r_{vi} — оценка пользователя v для того же товара.

User2user рекомендации

Как выбрать $s(u, v)$?

3. Корреляция Пирсона между векторами общих рейтингов

Позволяет определить направление и силу линейной зависимости между оценками пользователей:

$$s(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}$$

Где:

- \bar{r}_u — средний рейтинг, выставленный пользователем u .
- \bar{r}_v — средний рейтинг, выставленный пользователем v .

User2user рекомендации

Как выбрать $s(u, v)$?

4. Корреляция Пирсона с поправкой для малого числа общих рейтингов

Используется, если число общих оцененных товаров между пользователями мало. В этом случае вводится дополнительный коэффициент, уменьшающий влияние корреляции при недостатке данных:

$$s(u, v) = \min \left(\frac{|I_u \cap I_v|}{50}, 1 \right) \cdot \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}$$

Где множитель $\min \left(\frac{|I_u \cap I_v|}{50}, 1 \right)$ снижает вес корреляции при малом количестве общих оценок.

Item2item рекомендации

Идея

Для товаров, которые уже были оценены пользователем, находим наиболее похожие объекты и рекомендуем их.

Item2item рекомендации

Определение схожести между товарами

Похожесть между объектами вычисляется с использованием косинусного сходства, скорректированного с учетом среднего рейтинга пользователей.

Формально, схожесть между двумя товарами i и j определяется следующим образом:

$$s(i, j) = \frac{\sum_{u \in U_i \cap U_j} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} (r_{uj} - \bar{r}_u)^2}}$$

Где:

- U_i — множество пользователей, поставивших рейтинг товару i .
- U_j — множество пользователей, поставивших рейтинг товару j .
- r_{ui} — рейтинг, который пользователь u выставил товару i .
- \bar{r}_u — средний рейтинг, который пользователь u ставляет товарам.

Item2item рекомендации

Оценка рейтингов для рекомендаций

После вычисления схожести товаров, на основе истории взаимодействий пользователя можно предсказать его оценки для товаров, которые он еще не оценивал. Таким образом, можно выбрать наиболее релевантные объекты для рекомендации.

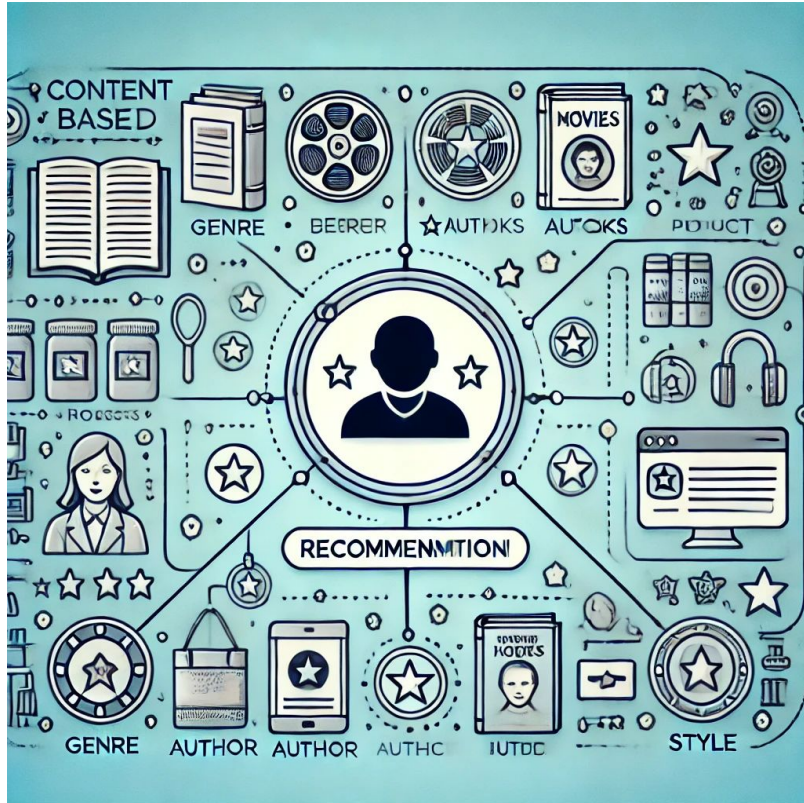
Особенности коллаборативной фильтрации

- Методы коллаборативной фильтрации не требуют дополнительной информации о пользователях и товарах, так как работают исключительно на основе истории взаимодействий.
- Неэффективны для новых пользователей и товаров с небольшим количеством данных (проблема холодного старта).
- Качество рекомендаций зависит от предыдущих взаимодействий, которые, в свою очередь, формируются на основе ранее предложенных рекомендаций.

План лекции:










- Постановка задачи
- Коллаборативная фильтрация
- Контентные рекомендации
- Рекомендательные системы на практике
- Особенности рекомендательных систем

Content-based рекомендации



Контентный подход основан на предложении пользователям объектов, схожих по характеристикам с теми, которые они уже оценили положительно.

Content-based рекомендации

User \ Item										
Петя										
Маша										
Вася										
Катя										

- Что порекомендовать Кате?
- Что лучше не рекомендовать Пете?

Content-based рекомендации

Основная идея

Каждому объекту i соответствует эмбединг e_i , который получен на основе его содержимого.

Для пользователя u можно рекомендовать объекты, схожие с уже оцененными, используя меру близости ρ между векторами эмбедингов.

Content-based рекомендации

Оценка рейтинга

Рейтинг объекта i для пользователя u можно вычислить как взвешенное значение рейтингов схожих объектов:

$$\hat{r}_{ui} = \max_{j \in I_u, r_{uj} > \alpha} \rho(e_i, e_j) r_{uj}$$

Здесь:

- I_u — множество объектов, которые пользователь уже оценил.
- $\rho(e_i, e_j)$ — мера схожести между объектами i и j .
- α — пороговое значение рейтинга.

Content-based рекомендации

Выбор меры схожести ρ

Мера схожести между эмбедингами объектов может быть различной:

1. Косинусное расстояние

Определяет угол между векторами эмбедингов:

$$\rho(e_i, e_j) = \frac{\langle e_i, e_j \rangle}{\|e_i\| \cdot \|e_j\|}$$

2. Скалярное произведение

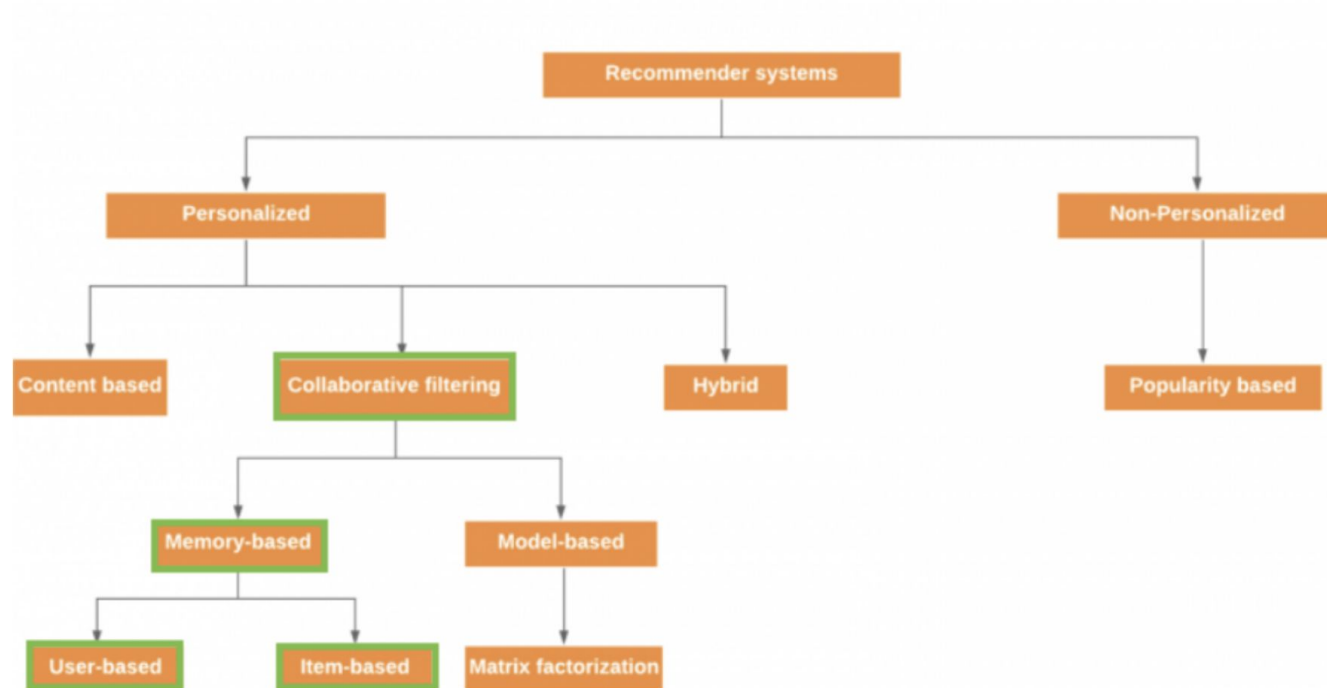
Оценивает степень совпадения значений эмбедингов:

$$\rho(e_i, e_j) = \sum_{k=1}^d e_{ik} e_{jk} = \langle e_i, e_j \rangle$$

3. Другие методы

Возможны и другие варианты метрик схожести, например евклидово расстояние, корреляция Пирсона и т. д.

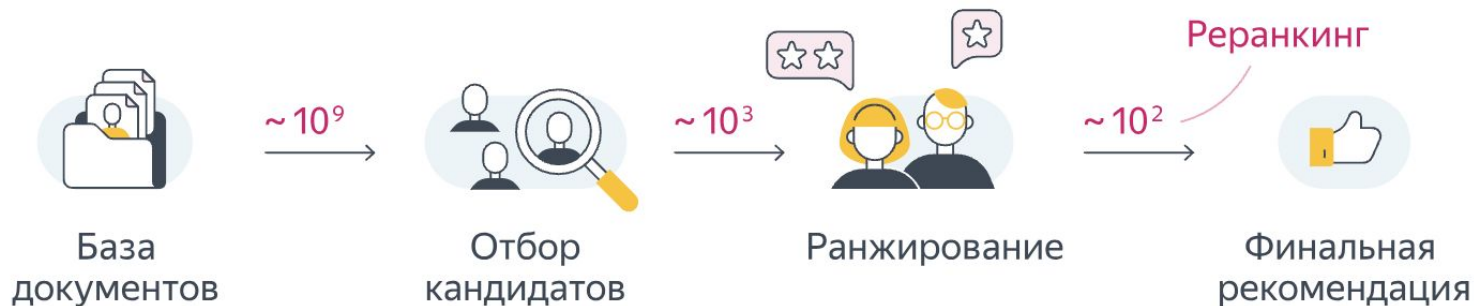
RecSys: taxonomy



План лекции:

- Постановка задачи
- Коллаборативная фильтрация
- Контентные рекомендации
- Рекомендательные системы на практике
- Особенности рекомендательных систем

Рекомендательные системы на практике



- Отбор кандидатов
- Ранжирующая модель
- Реранкинг

Отбор кандидатов

- В реальных рекомендательных системах число объектов может достигать сотен миллионов или даже миллиардов.
- Некоторые сложные модели слишком ресурсоемки, чтобы работать с таким объемом данных напрямую.
- При каждом запросе пользователя система должна обрабатывать все объекты, что делает вычисления неэффективными.

Решение:

На первом этапе следует отобрать ограниченное количество кандидатов, на которых затем будут применяться более сложные методы. Этот процесс называется **отбором кандидатов** и является первым шагом в пайплайне рекомендаций.

Отбор кандидатов

Подходы к отбору кандидатов

1. Эвристические методы

Отбор кандидатов на основе простых правил, таких как популярность товаров или их востребованность среди определенных групп пользователей.

2. Коллаборативные методы

Используют схожесть между пользователями или товарами, например, Item2Item или User2User подходы.

3. Контентные методы

Определяют схожесть объектов на основе эмбедингов, с возможностью ускоренного поиска с помощью структур данных, таких как HNSW и FAISS.

4. Методы с учетом бизнес-логики

Принимают во внимание факторы, влияющие на стратегию рекомендаций, например, свежесть и новизну товаров.

Рекомендательные системы на практике



- Отбор кандидатов
- Ранжирующая модель
- Реранкинг

Ранжирующая модель

1. Достоинства и недостатки моделей

Коллаборативные и контентные модели обладают как преимуществами, так и ограничениями, что делает их применение в чистом виде не всегда оптимальным.

2. Учет контекстной информации

Предпочтения пользователя могут меняться в зависимости от контекста, например, в разные дни недели или в разное время суток.

3. Использование данных о товарах

Дополнительная информация, такая как бренд, цена и другие характеристики, может значительно улучшить качество рекомендаций.

4. Комбинированный подход

На практике чаще всего применяется ранжирующая модель, объединяющая разные признаки для более точного формирования рекомендаций.

Ранжирующая модель

Признаки:

- Предсказание базовых коллаборативных, контентных моделей
- Признаки пользователя: пол, возраст, характеристики истории взаимодействия
- Признаки объекта: цена, вес, жанр, характеристики истории взаимодействий
- Контекстная информация: день недели, погода, местоположение

Задачи обучения:

- Бинарная классификация
- Регрессия
- Ранжирование

Модель:

Как правило, используется бустинг

Рекомендательные системы на практике



- Отбор кандидатов
- Ранжирующая модель
- Реранкинг

Реранкинг

На практике часто необходимо учитывать различные бизнес-ограничения, например, исключать устаревшие или слишком длинные видео, а также обеспечивать разнообразие рекомендаций.

Для этого после работы ранжирующей модели применяется **реранкинг** — механизм, который пересортировывает полученный топ объектов с учетом заданных условий

План лекции:

- Постановка задачи
- Коллаборативная фильтрация
- Контентные рекомендации
- Рекомендательные системы на практике
- Особенности рекомендательных систем

Особенности рекомендательных систем

- Проблема холодного старта
- Feedback loop

Проблема холодного старта

Проблема: как что-то порекомендовать новому пользователю? как понять, кому показывать новый товар?

Новый пользователь: чтобы рекомендовать что-то новому пользователю, нужно собрать максимум информации о нем. Это можно сделать с помощью технических методов или путем расширения регистрации и онбординга.

Новый товар: Для новых товаров, не имеющих истории взаимодействий, рекомендации можно строить на основе контентных моделей

Feedback loop

1. Рекомендательная система обучается на собственных рекомендациях, что может привести к локальному оптимуму и ограничить разнообразие.
2. Существует также риск, что она начнет преимущественно продвигать наиболее популярные товары или пользователей.

Простые способы избежать этого:

- Иногда добавлять случайные айтемы в рекомендации.
- Балансировать обучающую выборку по тегам, темам или популярности.