

Рекомендательные системы

SVD

Быковская Анна
Рудениев Алексей

Овчинников Павел
Чебаненко Дмитрий

Что такое рекомендательные системы и зачем они нужны?

Когда мы что-то ищем в интернете, будь то фильм, музыка или какой-то товар, мы склонны выбирать то, что нам предлагают **на основе наших предпочтений**, а точнее, что мы просматривали, оценивали или заказывали ранее. Например, когда пользователь посмотрел фильм в каком-нибудь онлайн-кинотеатре, у него обязательно появится карточка с рекомендациями, что-то вроде «Вам может это понравиться», «Рекомендации для вас» и прочее.

И в этом как раз помогают **рекомендательные системы** — это как личный помощник, который знает, что нравится пользователю, и предлагает именно то, что может его заинтересовать.

Где применяются?

- стриминговые сервисы (фильмы, музыка и др.)
- онлайн-кинотеатры
- онлайн-библиотеки
- социальные сети
- видеоконференции
- маркетинг
- медиа

Основная цель

Помощь пользователям в выборе товаров, услуг или контента, которые могут их заинтересовать

Рекомендательные системы

Что это?

Системы, которые **анализируют интересы** пользователей, чтобы представить пользователю или группе пользователей **персонализированные рекомендации**, которые могут быть им интересны. Выбранный алгоритм пытается **страграфизировать**, что ещё пользователю может понравиться.

Основные типы рекомендательных систем

Контентная фильтрация (Content-Based Filtering)
Этот метод учитывает сходство товаров, услуг или контента, который просматривает пользователь. Например, после просмотра фильма с определенным актером, система может порекомендовать пользователю другие фильмы, в которых снимался этот актер.

Коллаборативная фильтрация (Collaborative Filtering)
Делится ещё на 2 подвида фильтрации:
• User-based Collaborative Filtering
• Item-based Collaborative Filtering
Но каждый подвид метода учитывает поведение **нескольких** пользователей с похожими интересами. То есть система будет анализировать данные не о товаре, а о пользователе, который просматривает товар, и если есть другой пользователь с похожими предпочтениями, который смотрел другие модели с похожими характеристиками, то система будет их рекомендовать.

Матричные разложения (Matrix Factorization)
Использует методы факторизации матриц, например, **сингулярное разложение (SVD)**. Эти методы используют скрытые факторы для анализа взаимодействия пользователей и контента, что позволяет формировать более точные рекомендации.

Как работают рекомендательные системы



Что такое сингулярное разложение (SVD)?

В этом курсе практической линейной алгебры мы уже знакомились с **SVD-разложением**, но для более глубокого понимания материала давайте всё же вкратце вспомним, что это такое.

Сингулярное разложение — это представление прямоугольной матрицы $M_{m \times n}$ в виде произведения $U_{m \times m} \times \Sigma_{m \times n} \times V_{n \times n}^T$.

Где $U_{m \times m}$ и $V_{n \times n}^T$ — унитарные матрицы, состоящие из **левых** и **правых** сингулярных векторов соответственно, а $\Sigma_{m \times n}$ — диагональная матрица отсортированных по **убыванию** сингулярных чисел.

Удобство SVD заключается в том, что мы можем **сократить используемый объём данных**, при этом не потеряв основную часть оригинальной информации. Такое усеченное SVD называется **усечённым сингулярным разложением**.

Например, мы хотим сохранить только 90% информации. Для этого мы оставим r наибольших сингулярных чисел таких, что сумма квадратов этих элементов составит 90% от общей суммы квадратов диагональных чисел.

Тогда такое SVD будет иметь вид $M'_{m \times n} = U'_{m \times r} \times \Sigma'_{r \times r} \times V'^T_{r \times n}$. Матрица $M'_{m \times n}$ будет хорошо приближать исходную матрицу $M_{m \times n}$.

Усечённое сингулярное разложение используется для:

- скатия фотографий
- анализа текста
- составления рекомендаций
- и многое другое.

Мы же рассмотрим применение SVD в **рекомендательных системах**.

Как применяется SVD в рекомендательных системах?

Рекомендательные системы основываются на **предпочтениях конкретного пользователя**, поэтому матрица для сингулярного разложения называется **«user-item»**. То есть при построении матрицы на строках находятся **«пользователи»**, на столбцах **«предметы»**, а числом задается, например, **«отзывы»** пользователя на товар.

	Предмет 1	Предмет 2
Пользователь 1	5	3
Пользователь 2
...

Предположим, у нас есть данные о некотором количестве пользователей и их оценках на товары с AliExpress. Очевидно, так как не каждый пользователь заказывал тот или иной товар или просто не оставил отзыв, то пустые ячейки в матрице нужно заполнить нулями.

Придумаем таблицу

	Realme buds air 3	Комплект Xeon E5 2630v3	Видеокарта	Клавиатура «Чин Пын Гын»	Шнурки «Сунъыны»	Роскошь «Хлопушка»	Кронштейн «Три коня»	Амулет древних русов	Икона троицы	Сборник стихов Пушкина 69 томов
Овсянников П. А.	5	0	1	0	4	0	0	0	3	4
Зелепутер А. Ю.	0	2	5	0	0	0	0	0	2	0
Братишкін Н. И.	0	0	0	3	2	5	0	0	0	0
Перевозиков А. В.	0	3	5	0	0	0	0	2	1	0
Кудрявич Н. И.	1	1	5	0	0	4	4	0	5	1
Искандров А. Н.	3	0	3	0	4	0	2	0	0	2
Великов К. О.	5	0	0	0	3	0	0	0	4	5
Румянцев А. А.	0	4	0	5	0	5	0	5	0	0
Чебурков Д. А.	0	0	0	5	0	5	5	5	4	0

Составим по ней матрицу

[5 0 1 0 4 0 0 3 4 1]
[0 2 5 0 0 0 0 2 0 0]
[0 0 0 3 2 5 0 0 0 0]
[0 3 5 0 0 0 2 1 0 2]
[1 1 5 0 0 4 4 0 5 1]
[3 0 3 0 4 0 2 0 0 2]
[5 0 0 0 3 0 0 4 3 5]
[0 4 0 5 0 5 0 5 0 0]
[0 0 0 5 0 5 5 5 4 0]

Если мы **разложим** матрицу по SVD, урежем каждое составляющее произведение и после **перемножим**, то мы, конечно, получим **матрицу рейтингов**, однако из-за присутствия нулей в исходной матрице предсказание будет **неточным**, так как в некоторых ячейках получаются нули, а это, что всё-таки не будут заполнены нулями, вероятнее всего, будут иметь низкую оценку, что не очень похоже на реальную ситуацию.

Чтобы заполнить пустые ячейки, напишем **ML-модель**, которую обучим прогнозировать **примерную оценку** пользователя.

Каждый пользователь представлен в виде вектора соответствующей строкой в матрице P , а товары с AliExpress описываются вектором матрицы Q . Тогда такое взаимодействие представляется скалярным произведением $r_{uq} = \langle p_u, q_q \rangle$, где индексы u, q — порядковые номера пользователя и товара соответственно.

Обучить модель будем с помощью метода **градиентного спуска**.

Градиент — это направление роста скалярной величины. Сначала возьмём некоторую точку высоко на градиенте. Будем идти против градиента, чтобы найти **минимум** — итеративно вычислять функцию штрафа или же функцию ошибок, считать её градиент и обновлять параметры против градиента, чтобы **минимизировать ошибки**.

В качестве функции ошибки **LOSS** возьмём дисперсию — меру разброса значений случайной величины относительно её математического ожидания.

Усечённое сингулярное разложение используется для:

- скатия фотографий
- анализа текста
- составления рекомендаций
- и многое другое.

Мы же рассмотрим применение SVD в **рекомендательных системах**.

Преимущества и недостатки SVD в рекомендательных системах?

Рекомендательные системы основываются на **предпочтениях конкретного пользователя**, поэтому матрица для сингулярного разложения называется **«user-item»**. То есть при построении матрицы на строках находятся **«пользователи»**, на столбцах **«предметы»**, а числом задается, например, **«отзывы»** пользователя на товар.

	Предмет 1	Предмет 2
Пользователь 1	5	3
Пользователь 2
...

Предположим, у нас есть данные о некотором количестве пользователей и их оценках на товары с AliExpress. Очевидно, так как не каждый пользователь заказывал тот или иной товар или просто не оставил отзыва, то пустые ячейки в матрице нужно заполнить нулями.

Придумаем таблицу

	Realme buds air 3	Комплект Xeon E5 2630v3	Видеокарта	Клавиатура «Чин Пын Гын»	Шнурки «Сунъыны»	Роскошь «Хлопушка»	Кронштейн «Три коня»	Амулет древних русов	Икона троицы	Сборник стихов Пушкина 69 томов
Овсянников П. А.	5	0	1	0	4	0	0	0	3	4
Зелепутер А. Ю.	0	2	5	0	0	0	0	0	2	0
Братишкін Н. И.	0	0	0	3	2	5	0	0	0	0
Перевозиков А. В.	0	3	5	0	0	0	0	2	1	0
Кудрявич Н. И.	1	1	5	0	0	4	4	0	5	1
Искандров А. Н.	3	0	3	0	4	0	2	0	0	2
Великов К. О.	5	0	0	0	3	0	0	0	4	5
Румянцев А. А.	0	4	0	5	0	5	0	5	0	0
Чебурков Д. А.	0	0	0	5	0	5	5	5	4	0

Создадим нашу модель

Недостатки