

Современные подходы к построению рекомендательных систем

Алексей Чернобровов

консультант по Data Science

Матричная факторизация

Матричная факторизация

j

	1+1	Три мушкетёра	12 стульев	Легенд а №17
Алексей	10	9	1	7
Борис		9	2	
Вова	1		6	
Коля	3	<div>?</div>	4	10
Петя		1		
Юля			3	6

i

Матричная факторизация

u_i — «интересы пользователей»

v_j — «параметры фильмов»

$$\hat{x}_{ij} \approx \langle u_i v_j \rangle = \sum_{k=1}^K u_{ik} v_{jk}$$

Матричная факторизация

$$\hat{x}_{ij} = \langle u_i v_j \rangle$$

$$\sum_{i,j} (\langle u_i v_j \rangle - \hat{x}_{ij})^2 \rightarrow \min$$

Матричная факторизация

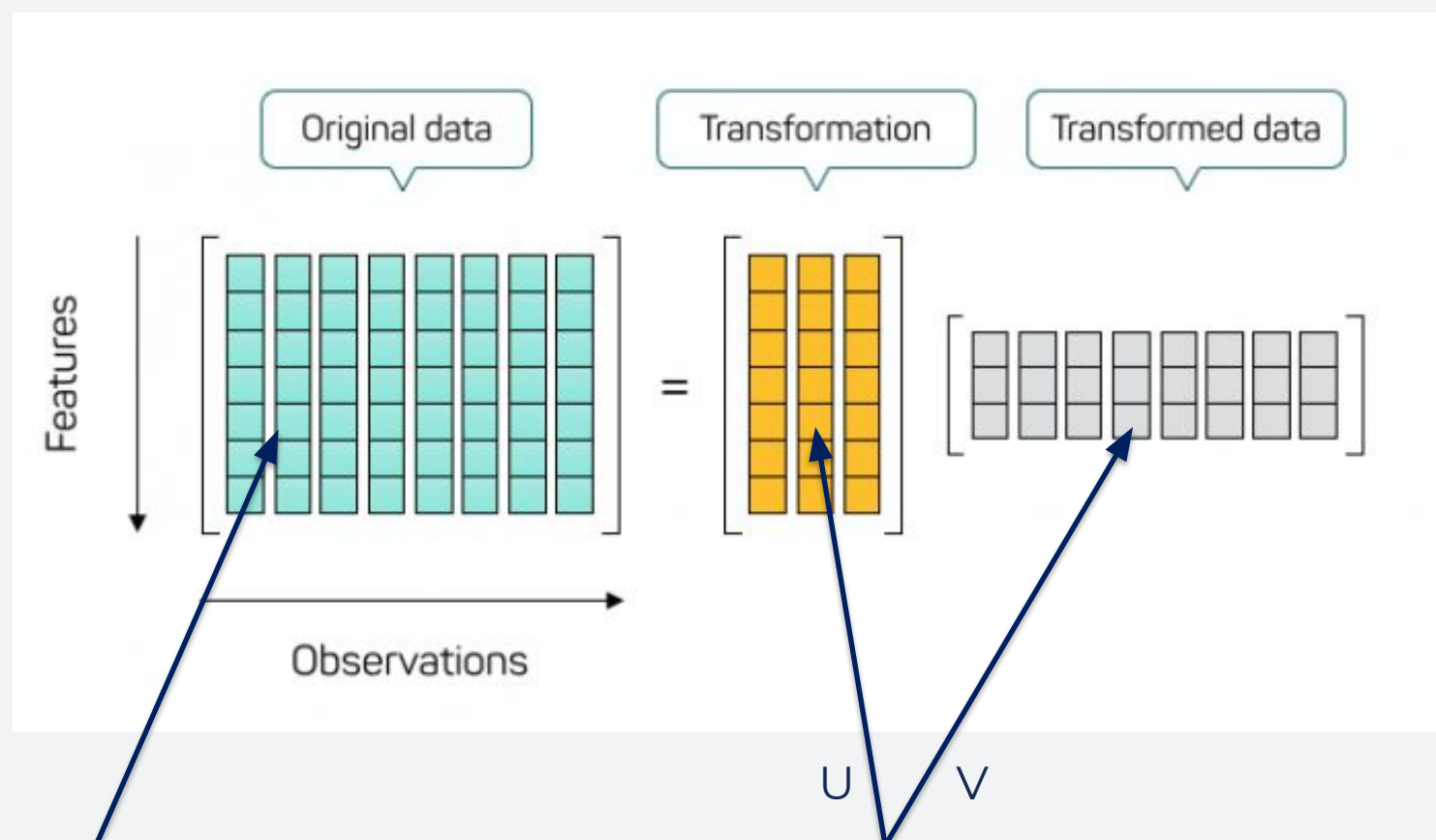
Факторизация — это операция разложения объекта (число, матрица) на его простые составляющие.

Способы разложения матрицы:

- PCA (Principal component analysis — метод главных компонент);
- SVD (Singular value decomposition — сингулярное разложение);
- NNMF/NMF (Non-negative matrix factorization — неотрицательное разложение матрицы).

Матричная факторизация

Идея факторизации матрицы:

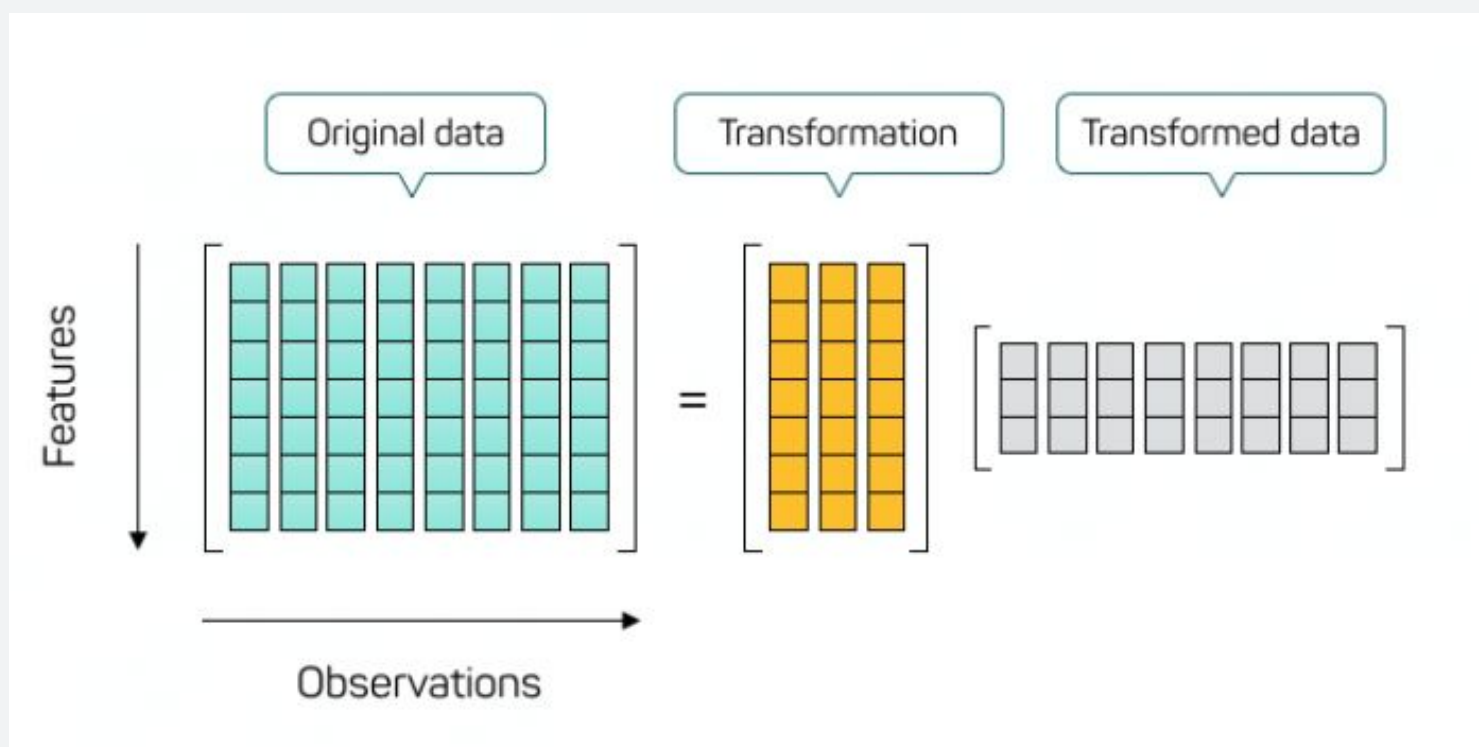


Исходная
матрица (X)

Две матрицы, на которые
мы хотим разложить
исходную

Матричная факторизация

Идея факторизации матрицы:



Общий вид разложения:

$X(m \times n) = U(m \times k) \cdot V(k \times n)$, где k — количество компонент.

Алгоритм SVD

SVD (Singular Value Decomposition)

переводится как сингулярное разложение матрицы.

- **Теорема о сингулярном разложении:**

у любой матрицы A размера $n \times m$ существует разложение в произведение трёх матриц: U , Σ и V^T :

$$\underset{n \times m}{A} = \underset{n \times n}{U} \times \underset{n \times m}{\Sigma} \times \underset{m \times m}{V^T}$$

- Матрицы U и V ортогональные, а Σ — диагональная (хотя и не квадратная).

$$UU^T = I_n, \quad VV^T = I_m,$$

$$\Sigma = \text{diag}(\lambda_1, \dots, \lambda_{\min(n,m)}), \quad \lambda_1 \geq \dots \geq \lambda_{\min(n,m)} \geq 0$$

Алгоритм SVD

SVD (Singular Value Decomposition)

переводится как сингулярное разложение матрицы.

- Помимо обычного разложения бывает ещё усечённое, когда из лямбд остаются только первые d чисел, а остальные мы полагаем равными нулю.

$$\lambda_{d+1}, \dots, \lambda_{\min(n,m)} := 0$$

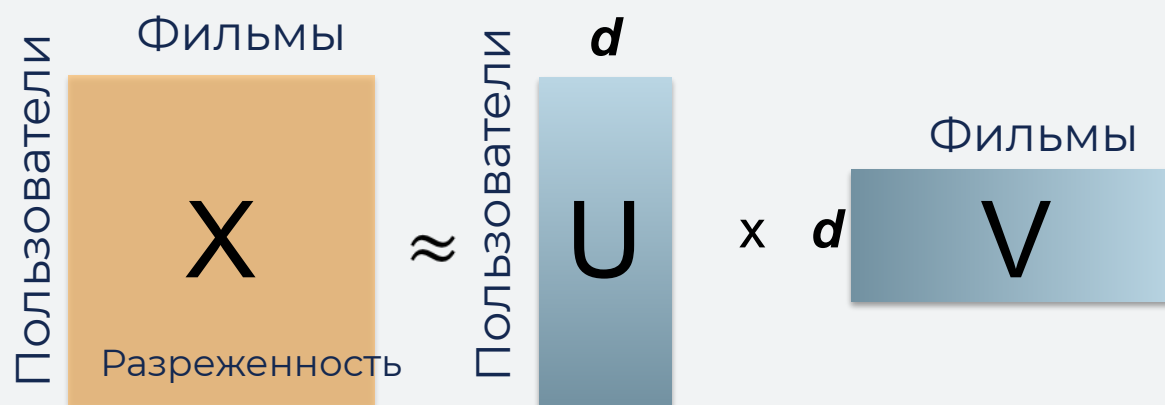
- Это равносильно тому, что у матриц U и V мы оставляем только первые d столбцов, а матрицу Σ обрезаем до квадратной $d \times d$.

$$\begin{matrix} A' & = & U' & \times & \Sigma' & \times & V'^T \\ n \times m & & n \times d & & d \times d & & d \times m \end{matrix}$$

SVD для рекомендаций

Чтобы предсказать оценку пользователя U для фильма V , мы берем некоторый вектор p_u (набор параметров) для данного пользователя и вектор для данного фильма q_i .

Их скалярное произведение и будет нужным нам предсказанием: $\hat{x}_{ij} = \langle u_i, v_j \rangle$



$$\hat{x}_{ui} = \langle u_i, v_j \rangle$$

Обобщение SVD

Саймон Фанк в статье в блоге, описывающей его решение Netflix Prize, предложил использовать более общий вид разложения:

$$\hat{x}_{ij} = \mu + b_i + b_j + q_j^T p_i$$

Особенности SVD

SVD хорошо работает с рейтингами.

Но если матрица задана бинарными значениями (0, 1), как, например, в случае, когда матрица заполняется покупками, SVD, как правило, показывает плохие результаты.

NMF (неотрицательное разложение)

Идея разложения сохраняется, однако в NMF применяются положительные матрицы.

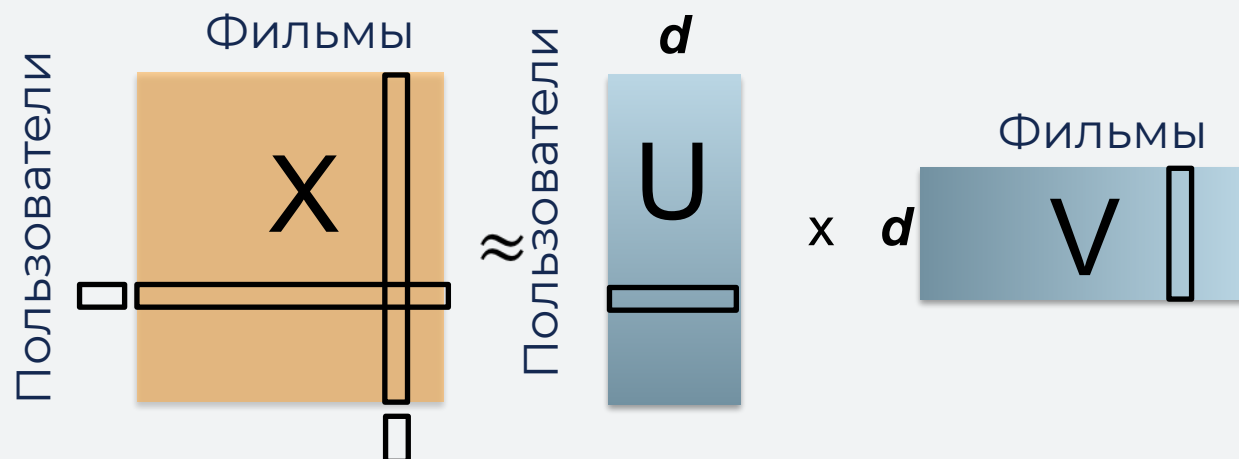
Недостатком NMF является, что в общем случае построения такое разложение является NP-трудной задачей.

ALS

ALS (alternating least squares) — популярный итеративный алгоритм разложения матрицы предпочтений на произведение двух матриц: факторов пользователей (U) и факторов товаров (V).

Работает по принципу минимизации среднеквадратичной ошибки на предоставленных рейтингах.

Оптимизация происходит поочерёдно, сначала по факторам пользователей, потом по факторам товаров. Также для обхода переобучения к среднеквадратичной ошибке добавляются регуляризационные коэффициенты.



Skillbox

Практика

Алексей Чернобровов

консультант по Data Science

Skillbox

Разбор практики

Алексей Чернобровов

консультант по Data Science

Проблема «ХОЛОДНОГО старта»

Алексей Чернобровов

консультант по Data Science

Проблема «холодного старта»

Новый пользователь ещё не успел проявить активность в системе, и о нём практически ничего не известно, а новый объект ещё никто не оценил. Возникает проблема «холодного старта».

Решение проблемы:

- Использовать средние оценки или взвешенное среднее всех «предсказанных» результатов и т.п.
- Использовать матричную факторизацию специального вида.
- Применяется принцип суперпозиции рекомендательных систем или **гибридные рекомендательные системы.**



Как валидировать проблему «холодного старта»?

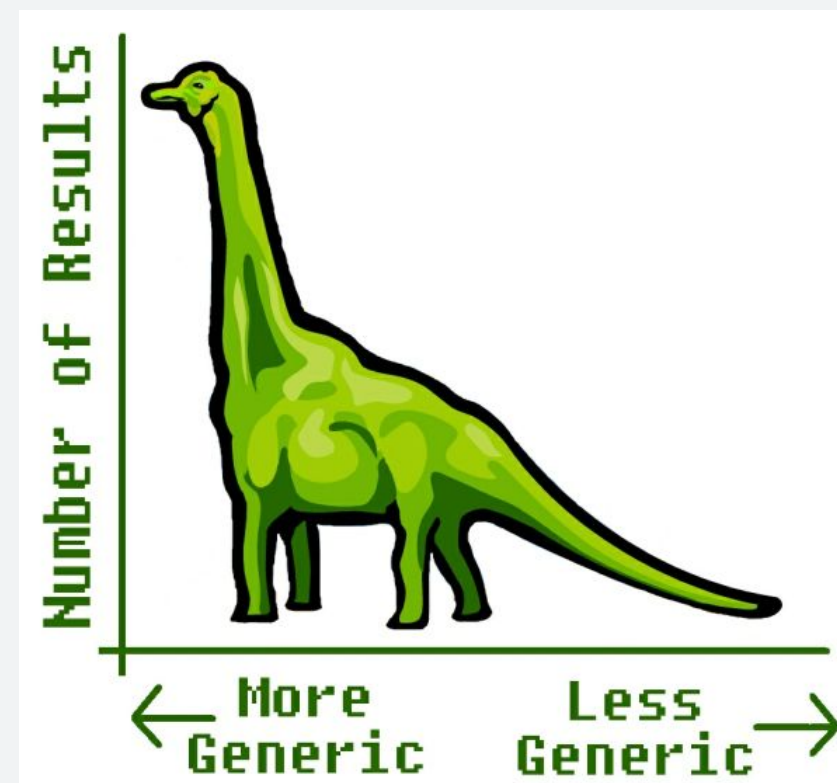
Валидировать проблему «холодного старта» оффлайн довольно сложно, поэтому самый простой вариант — её тестировать онлайн на реальных пользователях и смотреть, какой результат получится.



Проблема «длинного хвоста»

«Длинный хвост» — розничная концепция, описывающая явление больших суммарных продаж товаров, ставших в своё время классикой, по сравнению с товарами, которые в настоящее время считаются модными. Концепцию «Длинный хвост» впервые сформулировал Крис Андерсон.

Одно из решений — объединять товары, для которых мало статистики, в кластеры или в какую-то категорию.



Подходы к построению рекомендательных систем

Алексей Чернобровов

консультант по Data Science

Методы построения рекомендательных систем

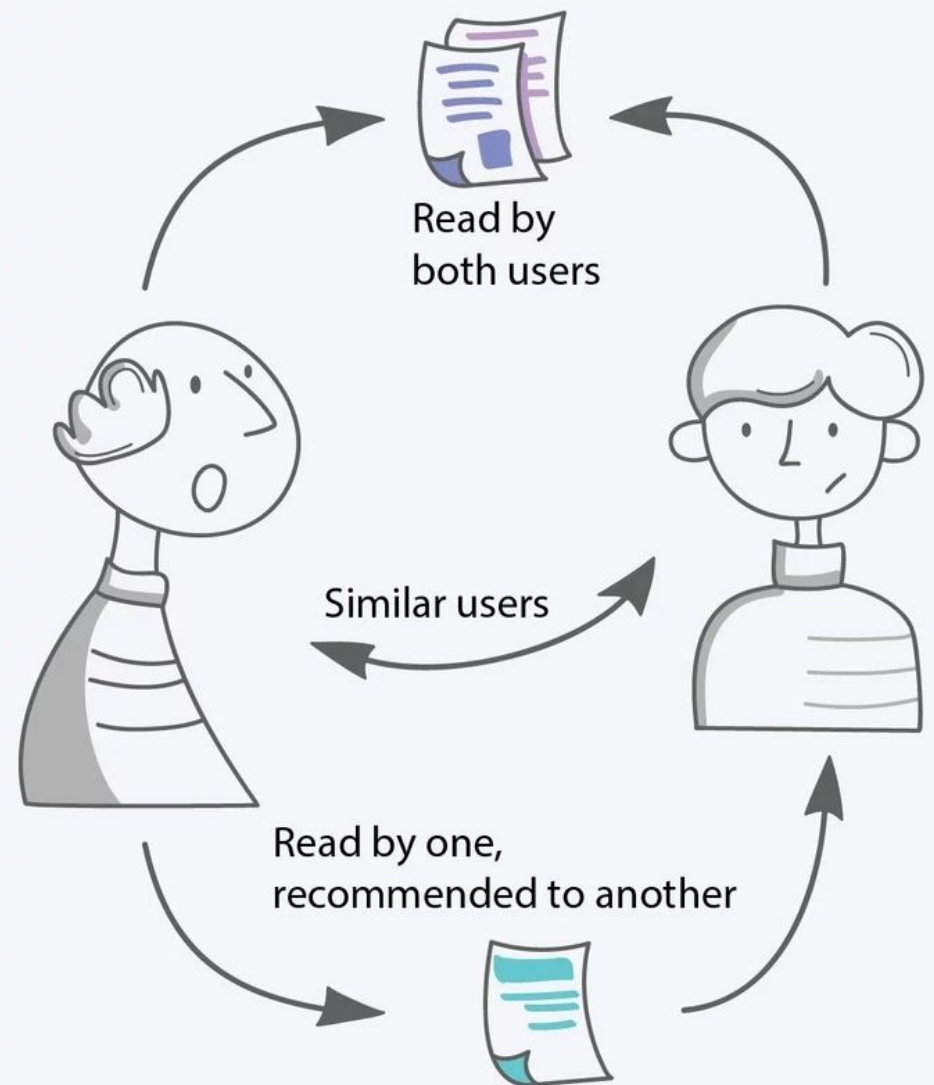
- Collaborative filtering.
- Content-based.
- Demographic.
- Utility-based.
- Knowledge-based.



Collaborative filtering

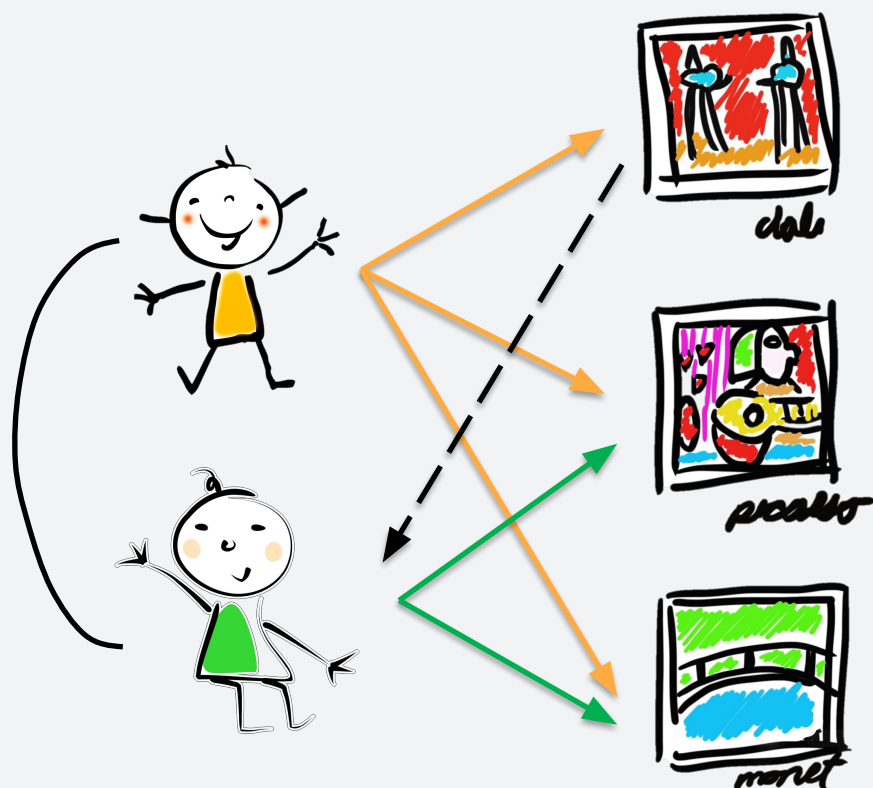
- Рекомендации для пользователя строятся на основе оценок похожих пользователей.

COLLABORATIVE FILTERING

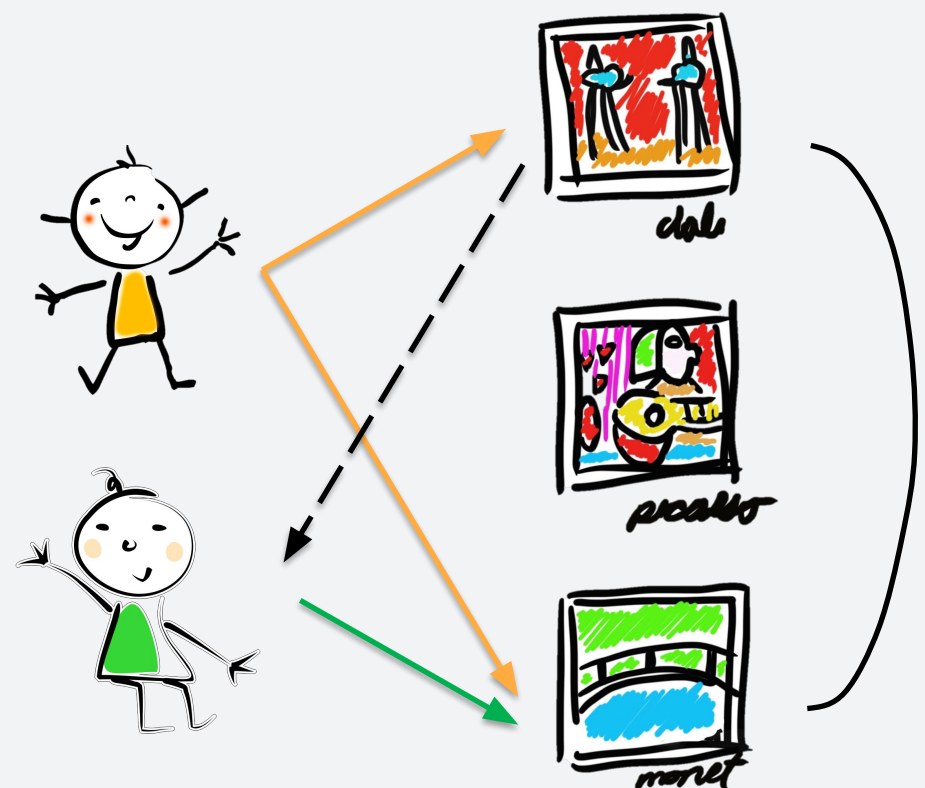


Collaborative filtering

1 User-based



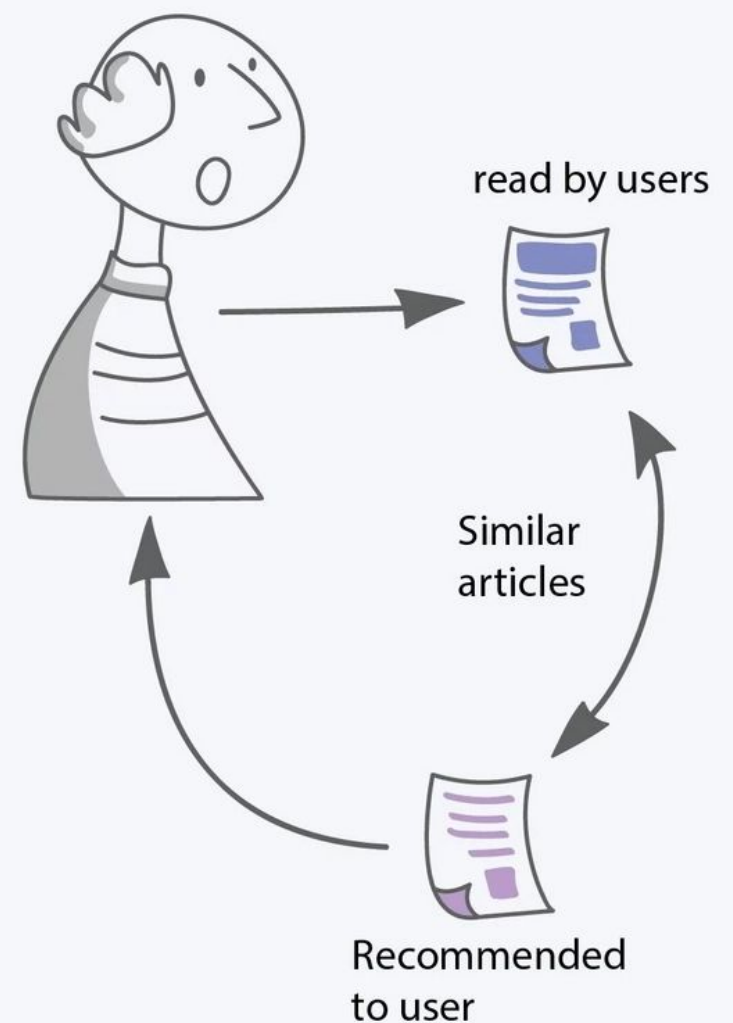
2 Item-based



Content-based

- Рассчитываются признаки для пользователей и объектов.
- Строится модель классификации/регрессии, приближающая оценки пользователей.

CONTENT-BASED FILTERING



Demographic

- Производится **сегментация пользователей** на группы.
- Рекомендации строятся на основе предпочтений группы.



Первое, что нужно сделать, — это собрать все данные, которые нам доступны о пользователе.

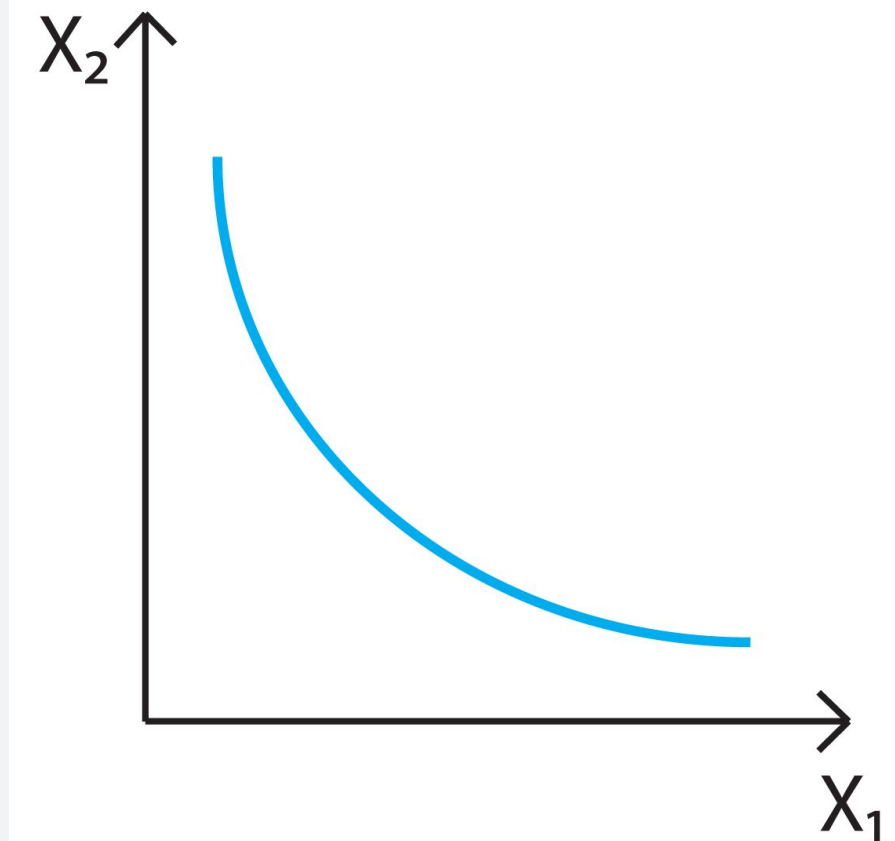
Например, нам могут быть известны социально-демографические характеристики пользователя: пол, возраст, род деятельности, образование. Далее необходимо провести сегментацию пользователей и посчитать предпочтения для полученных групп.

Рекомендация производится на основе того, в какую группу попадает новый пользователь.

Utility-based

Для каждого пользователя строится utility function, которая будет оценивать полезность данного объекта для данного пользователя.

Соответственно, такую utility function нужно рассчитать для каждого пользователя и далее на её основе строить рекомендации, выбирать только те объекты, которые получают большое значение по этой функции.



Knowledge-based

- Строится база знаний о том, как объекты *I* соотносятся с интересами и предпочтениями пользователя, которая с помощью правил эти соотношения описывает.
- Далее на основе предпочтений пользователей опять оценивается полезность объектов по этим правилам, и на основании этой полезности строятся рекомендации.



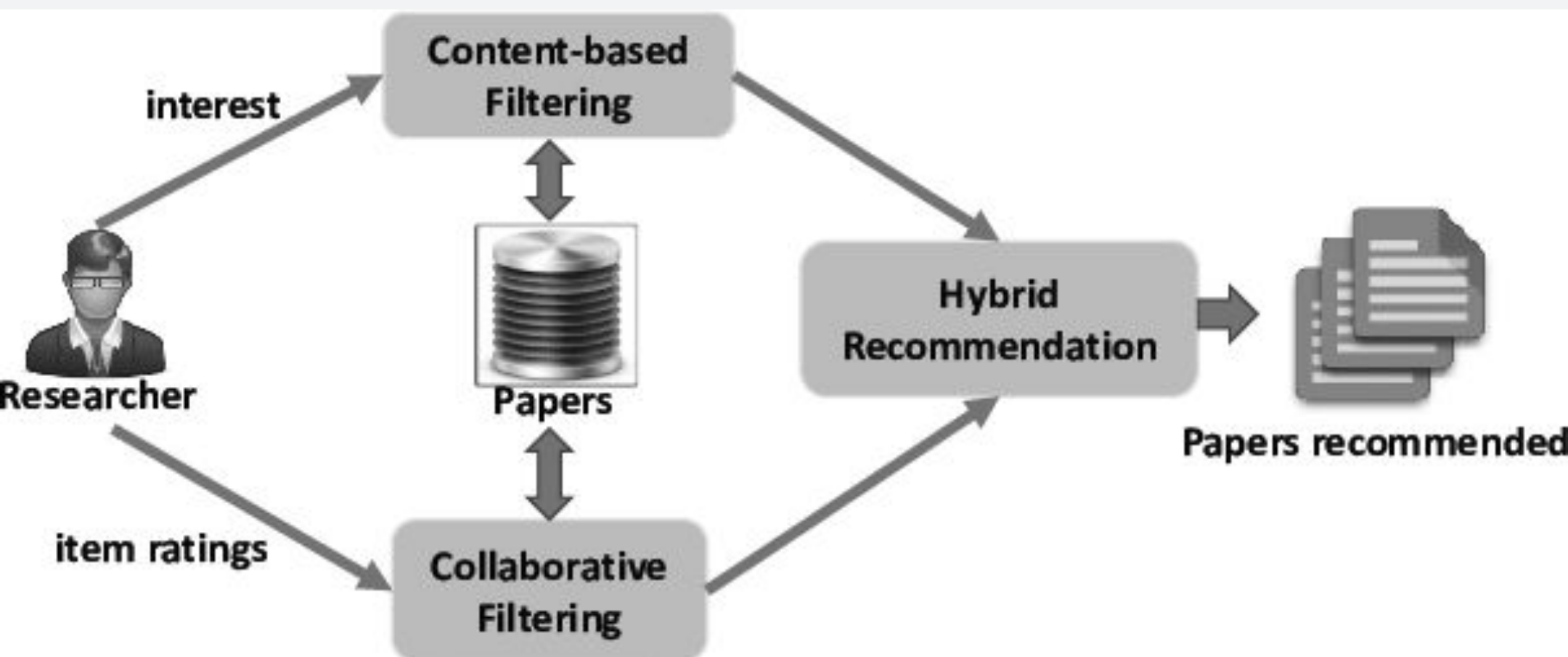
Skillbox

Гибридные ПОДХОДЫ

Алексей Чернобровов

консультант по Data Science

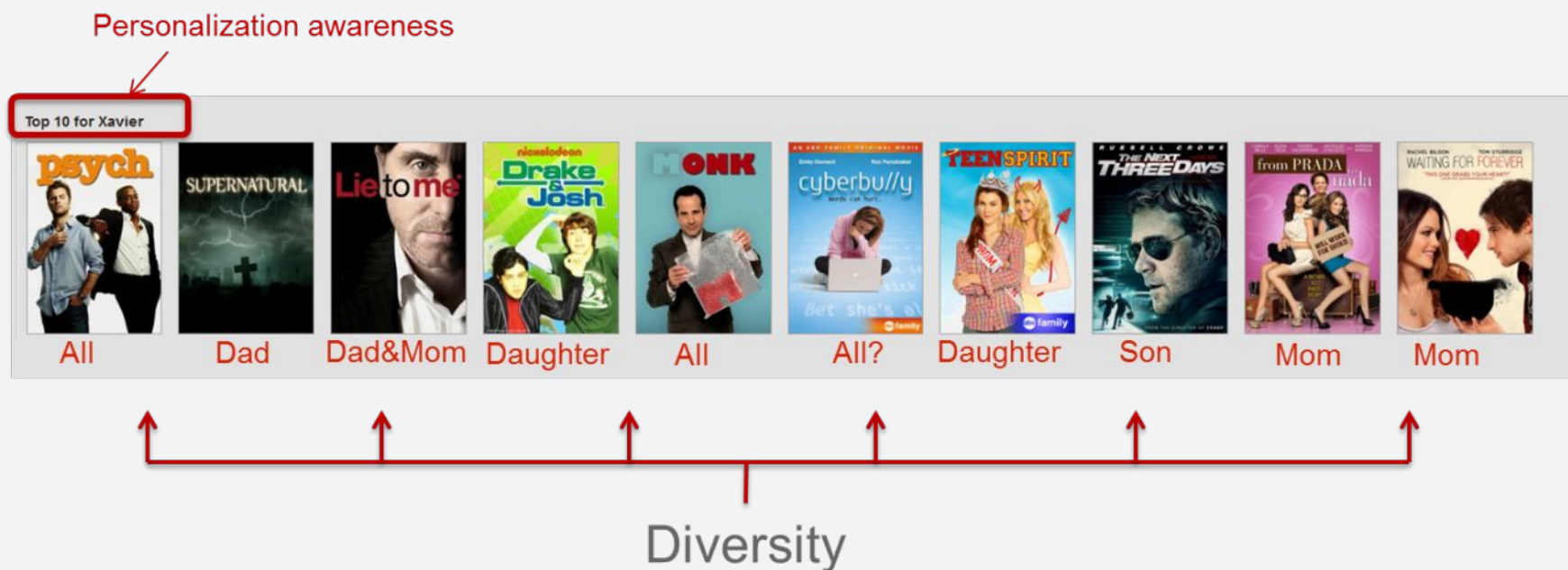
Гибридный подход



Гибридный подход

Гибридный подход — это сочетание коллаборативной и контентной фильтратий.

- Улучшает качество рекомендаций.
- Позволяет сначала взвешивать результаты согласно контентной фильтрации, а затем смещать эти веса по направлению к коллаборативной фильтрации (по мере «вызревания» доступного набора данных по конкретному пользователю).



Виды гибридизации

- Weighted,
- Switching,
- Mixed,
- Feature combination,
- Cascade,
- Feature augmentation.

Виды гибридизации

Weighted

- Рекомендации строятся на основе комбинирования оценок от разных систем с весами.
- Стратегия объединения: считать средневзвешенный прогноз по нескольким оценкам.
- Например:
 - линейная комбинация,
 - голосование.

Виды гибридизации

Switching

- Рекомендации строятся путём переключения между системами, работающими независимо, на основании критериев для переключения.
- Стратегия объединения: для разных продуктов/пользователей применять различные алгоритмы.

Виды гибридизации

Mixed

- Список рекомендаций состоит из «смеси» рекомендаций от разных систем.
- Стратегия объединения:
вычисляются рекомендации по разным алгоритмам, а потом просто объединяются в один список.

Виды гибридизации

Feature combination

- Подход основан на content-based.
- Признаки от разных систем объединяются в одну выборку для построения единой модели.

Виды гибридизации

Cascade

- Поэтапное применение нескольких моделей для уточнений рекомендаций.
- Candidate selection.

Виды гибридизации

Feature augmentation

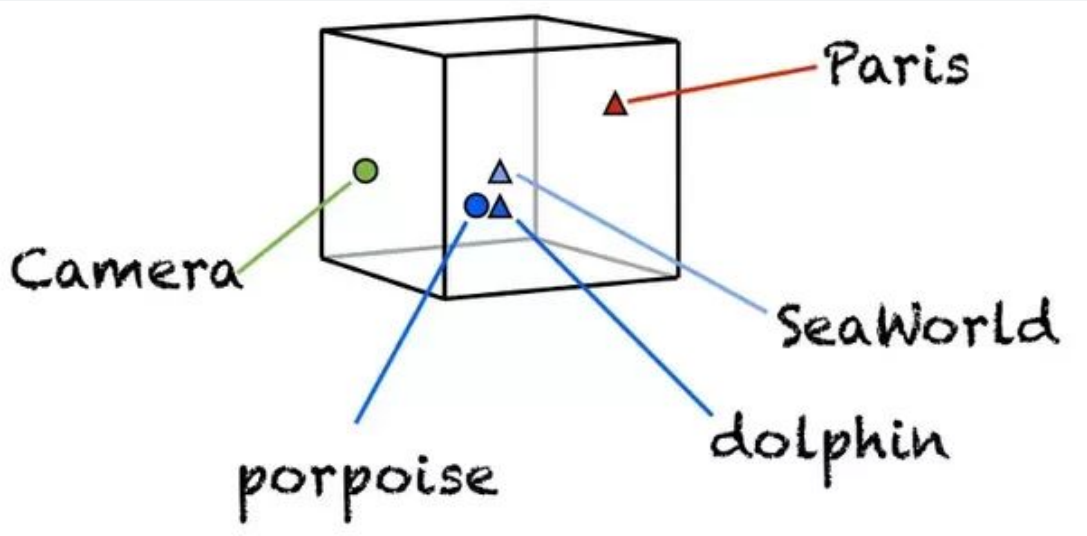
- Выход от одной или нескольких рекомендательных систем используется как входные признаки для другой системы.

Нейронные сети для задач рекомендаций

Embedding (векторное представление)

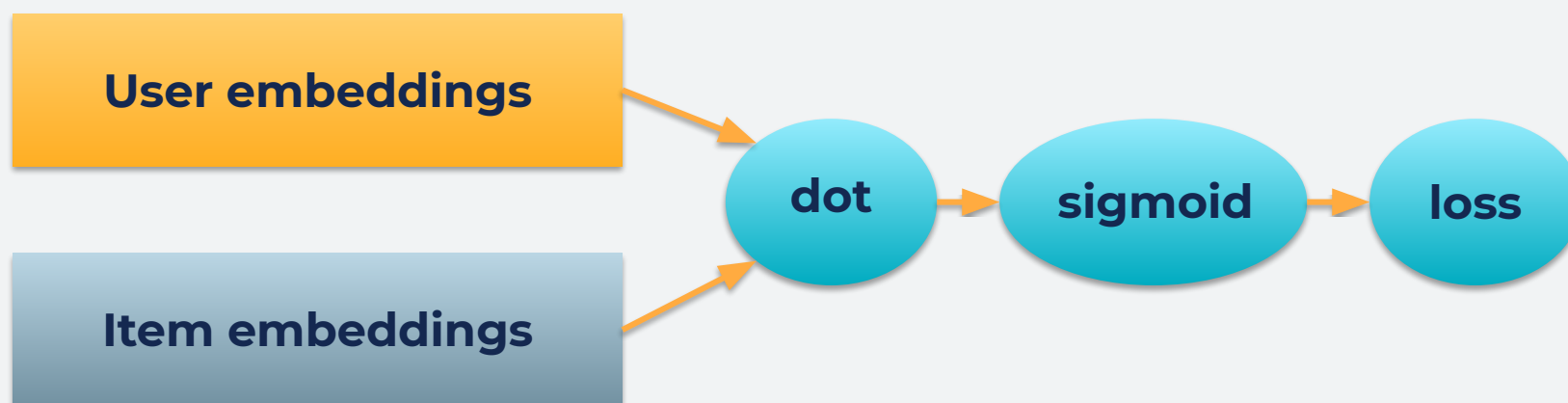
Embedding — это сопоставление произвольной сущности (пользователя, фильма, текста) некоторому вектору.

	dim-0	dim-1	dim-2	dim-3	dim-4	...	dim-45	dim-46	dim-47	dim-48	dim-49
title											
War and Peace	-0.279165	-0.107367	0.114153	0.143709	-0.141921	...	-0.067178	0.230711	-0.230550	0.199285	-0.099167
Anna Karenina	-0.248443	-0.000578	0.150472	0.151845	0.000908	...	-0.141615	0.178011	-0.230794	0.042102	-0.189196
The Hitchhiker's Guide to the Galaxy (novel)	-0.190761	-0.060406	0.115548	-0.249868	-0.120824	...	-0.038944	0.084992	-0.047035	-0.054157	-0.209883



Дополнительные преимущества нейронных сетей

Использование



Обратная сторона

Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

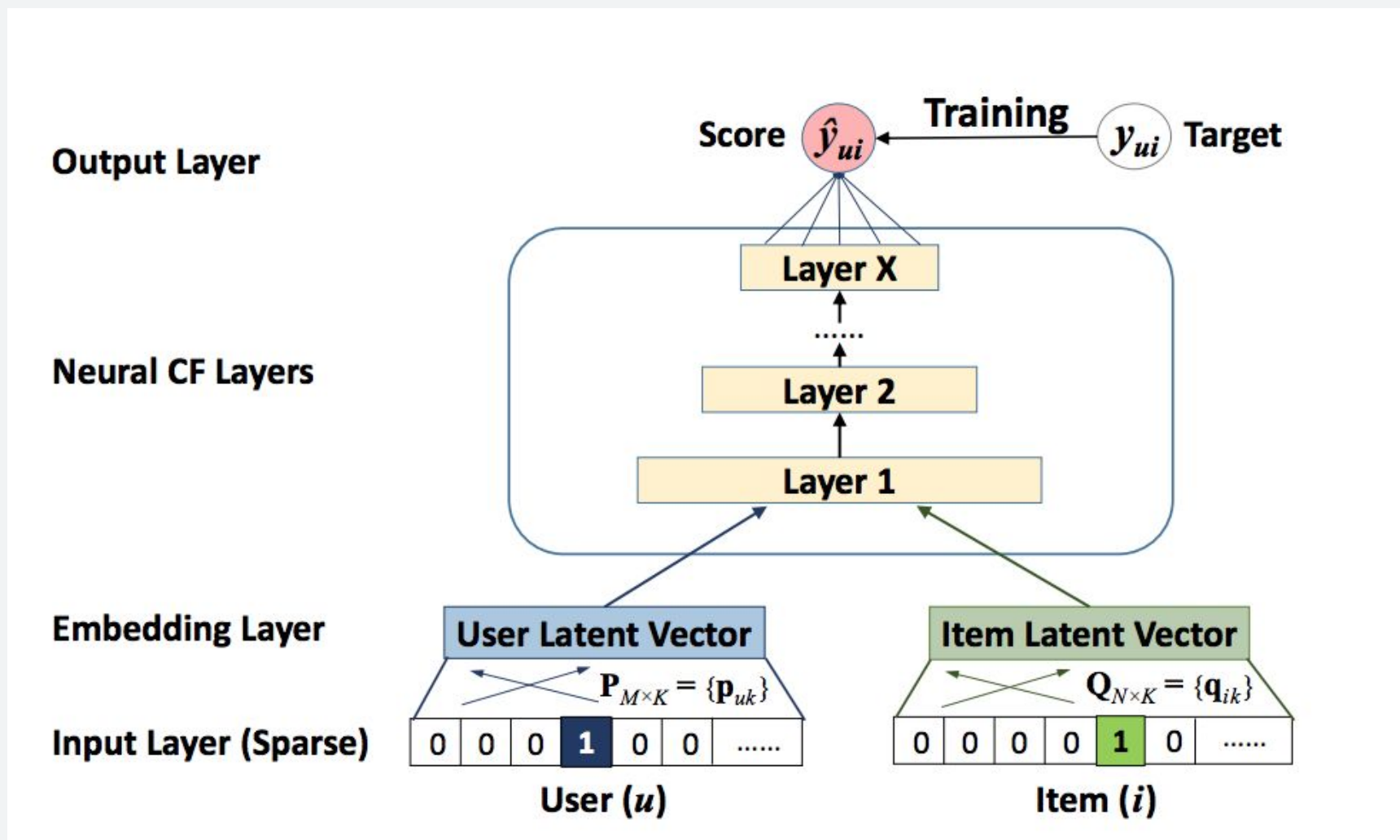
https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation

<https://arxiv.org/pdf/1907.06902.pdf>

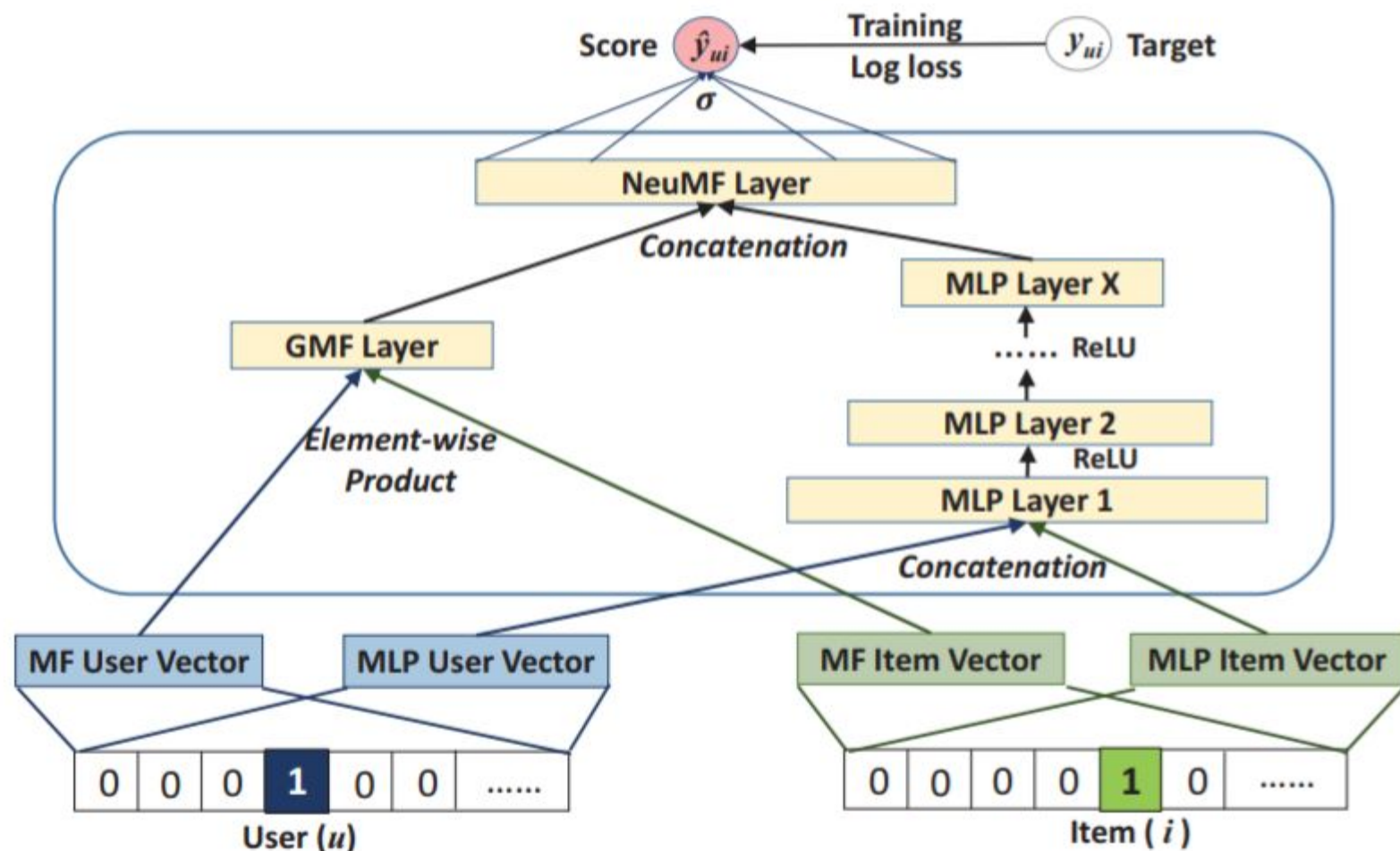
Skillbox

Практика

Neural Collaborative Filtering



Neural Collaborative Filtering



Как развиваются рекомендательные системы?

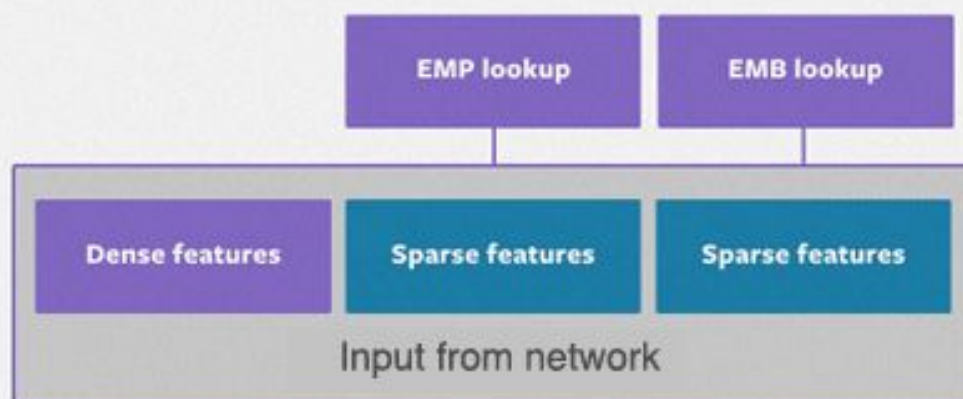
Развитие в рекомендательных системах

Развитие рекомендательных систем и их усовершенствование идёт постоянно — и идёт по следующим направлениям:

- количество контента, который мы рекомендуем;
- информация, которую мы знаем о контенте;
- информация о юзерах;
- интерфейсные решения.

Библиотека DLRM от Facebook

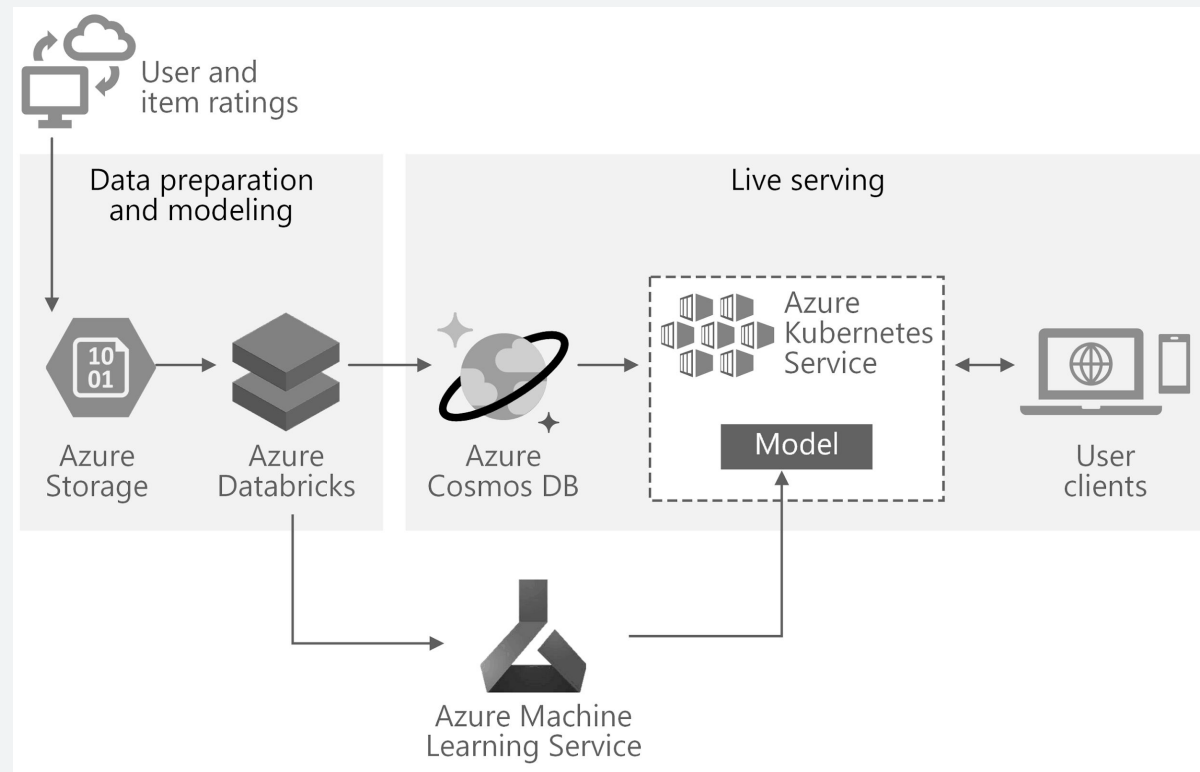
Библиотека DLRM на Python для построения рекомендательных систем.



Recommenders от Microsoft

<https://github.com/microsoft/recommenders>

- Десятки разных алгоритмов, которые постоянно пополняются.
- Встроенные dataset.
- Удобство интеграции с Azure.



Заключение

Что мы рассмотрели:

- гибридный подход и виды гибридизации,
- матричную факторизацию,
- нейронные сети для задач рекомендаций.

**Спасибо за
внимание!**