9.3. Контентные рекомендации

Введение

Все рекомендательные системы можно разделить на три типа: коллаборативные, контентные и гибридные.

Основная идея контентных рекомендаций в том, что для их построения будут использоваться атрибуты юзеров и товаров.

Какими бывают контентные признаки

Признаки товара на примере музыкального сервиса:

- Стандартные статистики объекта: количество лайков, кликов, полных прослушиваний
- Признаки автора: количество слушателей, жанр
- Неструктурированные данные: названия треков, обложки или даже предобученные эмбединги треков

Признаки юзера на примере музыкального сервиса:

- Доступная информация: возраст, пол
- Контекст запроса: с какого устройства, в какое время
- Друзья юзера: усредн1нный эмбеддинг всех треков, которые слушал каждый из друзей

Факторизационные машины

Пусть I - множество товаров, U - множество юзеров. Для каждой пары товар-юзер построим вектор размерности $|U| + |I|$ взаимодействия этой пары, в котором единицы стоят на месте соответствующих юзера и товара.

	u_1	u_2	u_3	i_1	i_2	i_3	y
x_1	1	0	0	1	0	0	2
x_2	1	0	0	0	1	0	4
x_3	0	1	0	0	1	0	1
x_4	0	0	1	1	0	0	3
x_5	0	0	1	0	0	1	5
	users			items			observed ratings

Предсказывать будем юзерские рейтинги товаров $a(x)$.

Простейшая регрессионная модель: $a(x) = w_0 + \sum_{t=1}^{|U|+|I|} w_t x_t$
Заметим, что к этой табличке можно добавить любые фичи юзеров, товаров или их взаимодействия.

Factorization machine training data

	u_1	u_2	u_3	i_1	i_2	i_3	a_1	a_2	y
x_1	1	0	0	1	0	0	2,0	0,0	2
x_2	1	0	0	0	1	0	1,5	0,5	4
x_3	0	1	0	0	1	0	1,0	0,0	1
x_4	0	0	1	1	0	0	0,3	0,7	3
x_5	0	0	1	0	0	1	3,2	1,7	5
	users			items			auxiliary features		observed ratings

Дальше можно добавить фичи взаимодействия второго порядка:

$$a(x) = w_0 + \sum_{t=1}^{n} w_t x_t + \sum_{r=1}^{n} \sum_{s=r+1}^{n} w_{rs} x_r x_s$$

Всего фичей получается $\frac{n(n+1)}{2} + n + 1$, а это затруднительно для подсчёта.

Решить это можно заменой матрицы W на её низкоранговое приближение $V^T V$, где V - матрица с n x k векторами v_i по столбцам. Число фичей можно снизить до $nk + n + 1$.
Данная модель называется факторизационной машиной.

Примеры применения:

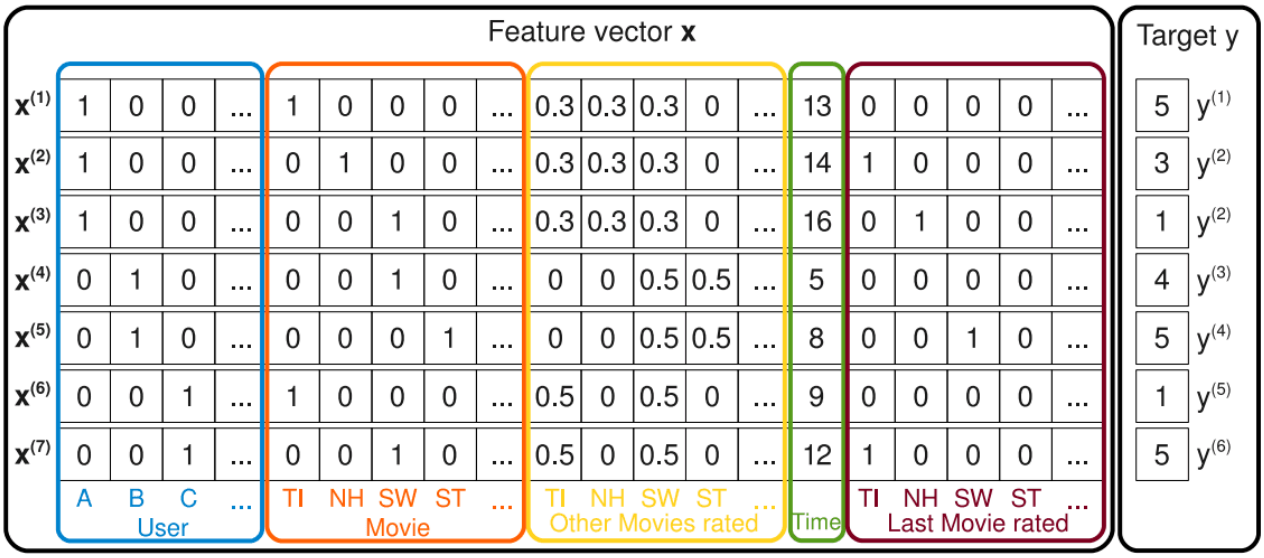
- Предсказание рейтинга, выход модели можно интерепретировать, как оценку
- Бинарная классификация, навесить логит на выход модели
- Можно ранжировать объекты

Модель обычно обучается градиентным спуском.

FFM – Field-aware Factorization Machines\

Идея в том, чтобы иметь несколько различных латентных представлений для каждой из фичей. То есть, когда раньше 1 фича комбинировалась 2 другими, то использовался 1 вектор первой фичи для обоих случаев. А теперь, для каждой комбинации будут использовать свои вектора. То есть модель будет вида:

$$a(x) = w_0 + \sum_{t=1}^{n} w_t x_t + \sum_{r=1}^{n} \sum_{s=r+1}^{m} < v_{r,s}, v_{s,r} > x_r x_s$$

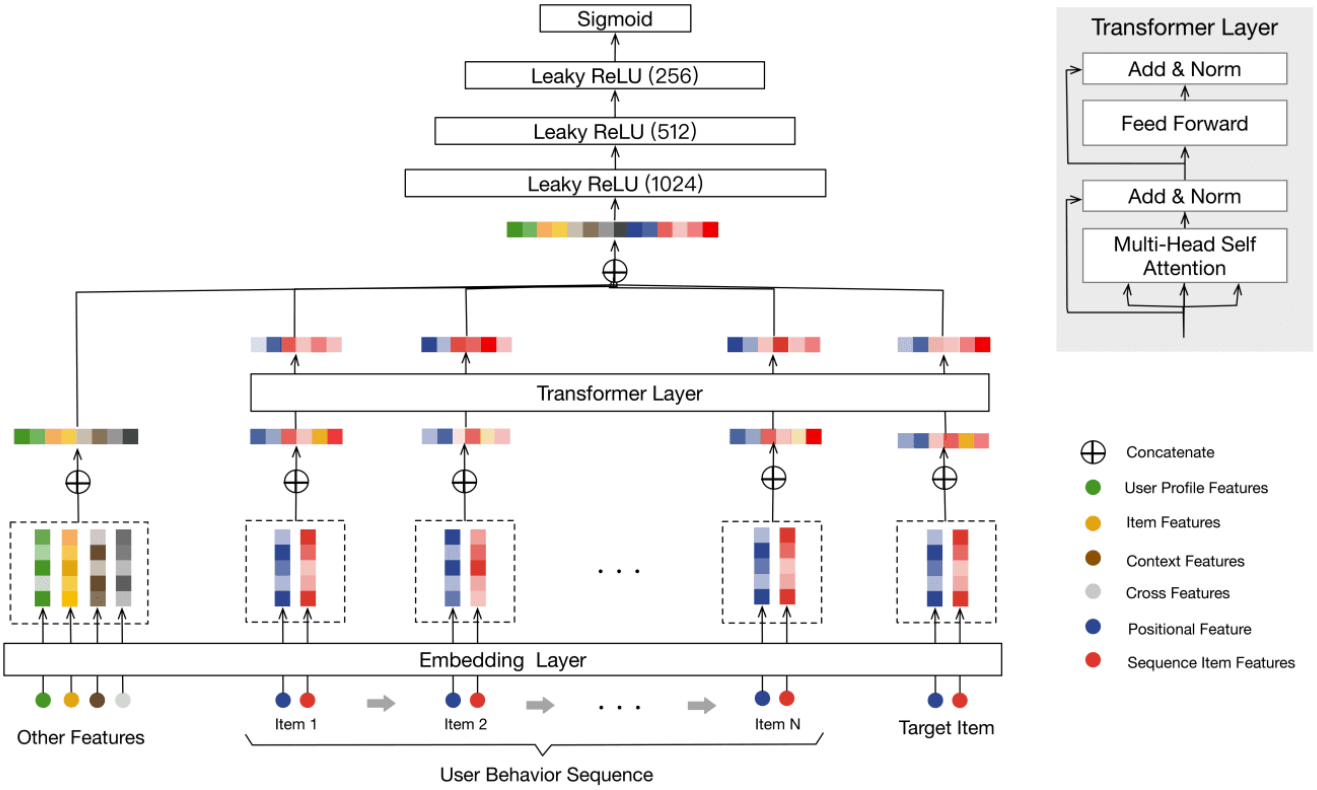


DSSM (deep sematic similiarity model)

Изначальная идея в том, что у нас есть текст запросы и текст документа, мы строим контентные эмбединги (с помощью нейронки) текстов и с помощью косинусной меры оцениваем схожесть. Эмбеддинги получаются так: по каждому тексту составляются триграммы (последовательность 3 подряд идущих слов), из них строится мешок (сортируются по частоте), затем пропускаем его через несколько полносвязных слоёв, максимизируя условную вероятность документа при условии запроса. Можно подобрать разные функции потерь.

Трансформеры для рекомендаций

В качестве последовательности токенов берём историю событий юзера, где каждый элемент это, например, клик на товар. Трансформемы позволяют учитывать порядок событий, сложные паттерны в поведении и интересах юзера.



На вход модели подается история кликов пользователя, на основе которой нужно предсказать вероятность клика по заданному объекту. Роль архитектуры трансформера здесь в том, чтобы качественно закодировать представление пользователя, после чего применяется обычный multi layer perceptron (MLP) для предсказания вероятности.

Помимо архитектур, которые специально разрабатываются под задачи рекомендаций, трансформеры можно использовать и как обособленные предобученные модели для построения векторых представлений текстов или изображений, которые затем подаются как признаки для решения downstream задач в домене рекомендаций. Несмотря на очевидные преимущества трансформеров с точки зрения качества, их использование в продакшене часто ограничивается имеющимися вычислительными ресурсами. Это особенно актуально для рекомендаций, где модели важно применять непосредственно в момент запроса пользователя.