Различные методы построения рекомендательных систем

Георгий Демин 417гр

ВМК МГУ

25.11.2019

Постановка задачи

Дано:

1. Множество пользователей $\mathcal{U}$, множество товаров $\mathcal{I}$ (возможно, с признаковым описанием)
   $|\mathcal{U}| = n$, $|\mathcal{I}| = m$
2. Контекст (время, место и т.д.)
3. Матрица интеракций (рейтингов) $\mathcal{R} \in \mathbb{Z}_+^{n \times m}$ рейтинги могут быть явными (explicit) и неявными (implicit)

	Товар 1	Товар 2	Товар 3	Товар 4	Товар 5
Клиент 1		3		5	
Клиент 2	1		1	1	
Клиент 3	2			3	2
Клиент 4		4			5
Клиент 5	5		2	3	4

Рис.: пример матрицы интеракции

Постановка задачи

Требуется найти функцию

$$f : (\mathcal{U} \times \mathcal{I}) \to \mathcal{R}$$

Стандартные обозначения:
$r_{ui} = R[u, i]$ - известная оценка
$\hat{r_{ui}} = f(u, i)$ - предсказанная оценка
U_i - множество пользователей, оценивших товар i
I_u - множество товаров, оценённых пользователем u
Общий подход:

$$\mathcal{L}(r_{ui} - \hat{r_{ui}}) \to \min_f$$

Основные подходы

1. Collaborative filtering
   1. Memory-based
   2. Model-based (matrix factorization)
2. Content-based filtering
3. Context-aware collaborative filtering

Memory-based

Похожим пользователям нравятся похожие товары:

$$\hat{r_{u,i}} = \frac{1}{|U_i|} \sum_{u' \in U_i} \text{sim}(u, u') r_{u',i}$$

Memory-based

Похожим пользователям нравятся похожие товары:

$$\hat{r_{u,i}} = \frac{1}{|U_i|} \sum_{u' \in U_i} \text{sim}(u, u') r_{u',i}$$

Добавляем среднее

$$\hat{r_{u,i}} = \mu + \bar{r_u} + \frac{1}{|U_i|} \sum_{u' \in U_i} \text{sim}(u, u')(r_{u',i} - \bar{r_{u'}} - \mu)$$

Memory-based

Преимущества:

1. Быстро
2. Просо

Недостатки:

1. Модель слишком простая
2. Зачастую рекомендуем самые популярные товары
3. Не можем учитывать признаки пользователей и товаров
4. Проблема холодного старта (cold-start)

Model-based

Основная идея - matrix factorization: разложить матрицу $\mathcal{R}$ в произведение матриц меньшего ранга
Похоже на SVD decomposition:
$R \in \mathbb{R}^{n \times m} = P \Sigma Q^T$, $\text{rank}(R) = k$
$P \in \mathbb{R}^{n \times k}$, $Q \in \mathbb{R}^{k \times m}$, $\Sigma = \text{diag}\{\sigma_i\}$, $\sigma_i \geqslant 0$

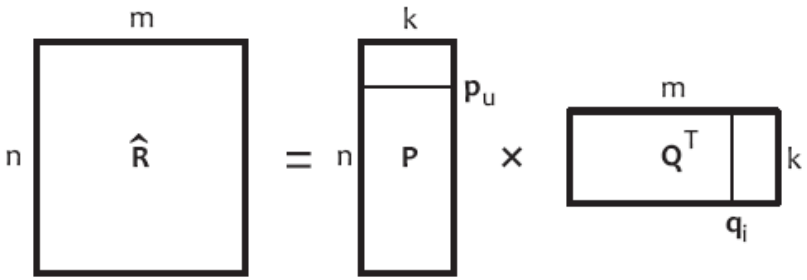

Рис.: matrix factorization

По сути учим k-размерные эмбединги для пользователей и товаров

Model-Based

Алгоритм **Alternating Least Squares**:

$$\sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\|x_u\|^2 + \|y_u\|^2) \to \min_{x,y}$$

$$p_{ui} = \begin{cases} 1 & \text{if } r_{ui} > 0 \\ 0 & \text{if } r_{ui} = 0 \end{cases}$$

$$c_{ui} = 1 + \alpha r_{ui}$$

Это функция потерь Ridge-регрессии, известен аналитический ответ
Оптимизировать можно блочно-координатным спуском

Model-Based

Преимущества **ALS**:

1. Получаем эмбединги пользователей и товаро
2. Простая оптимизационная задача
3. Неявно учитываем схожесть пользователей

Model-Based

Преимущества **ALS**:

1. Получаем эмбединги пользователей и товаро
2. Простая оптимизационная задача
3. Неявно учитываем схожесть пользователей

Недостатки:

1. Модель все ещё довольно простая
2. Зачастую рекомендуем самые популярные товары
3. Не можем учитывать признаки пользователей и товаров
4. Проблема холодного старта (cold-start)
5. При новых интеракциях нужно переучивать

Признаковое описание

Основная идея: будем использовать не только оценки товаров пользователями, но и дополнительную информацию.
Теперь $U \in \mathbb{R}^{n \times f_1}$, $I \in \mathbb{R}^{n \times f_2}$
Признаки пользователей: социально-демографические, связи в социальных сетях
Признаки товаров: цена, различные статистики, особенности из предметной области

Алгоритм LightFM

Каждый **признак** и пользователя, и товара кодируется вектором размерности k, то есть они лежат в одном пространстве.
e_s^u- эмбединг фичи пользователя с номером s
Эмбединги пользователей $q \in \mathbb{R}^{n \times k}$ и товаров $p \in \mathbb{R}^{m \times k}$ - сумма эмбедингов признаков.

$$q_j = \sum_{s=1}^{f_1} U[j, s] e_s^u \in \mathbb{R}^k$$

$$p_j = \sum_{s=1}^{f_2} I[j, s] e_s^i \in \mathbb{R}^k$$

Алгоритм LightFM

Предсказания:

$$\hat{r_{u,i}} = f(q_u p_i + b_u + b_i)$$

В качестве f можно брать разные неубывающие функции с областью значеиний - возможными r_{ui}. Далее, как и в статье будем рассматривать случай, когда ответ бинарный. В качестве f берём сигмоиду.

$$\mathcal{L}(r_{ui} - \hat{r_{ui}}) = \prod_{u,i}(r_{ui} - \hat{r_{ui}})$$

Алгоритм LightFM

Преимущества **ALS**:

1. Получаем эмбединги пользователей и товаро
2. Можем использовать внешнюю информацию
3. Лучшее представление пользователей и товаров
4. Отчасти решается проблема холодного старта

Алгоритм LightFM

Преимущества **ALS**:

1. Получаем эмбединги пользователей и товаро
2. Можем использовать внешнюю информацию
3. Лучшее представление пользователей и товаров
4. Отчасти решается проблема холодного старта

Недостатки:

1. Используется только линейная связь признаков
2. Зачастую рекомендуем самые популярные товары
3. При добавлении новых пользователей/товаров нужно переучивать
4. Очень долго учиться на dense-признаках

SLAGR

Social Influence-based Attentive Mavens Mining and Aggregative Representation Learning for Group Recommendation
Теперь у нас есть ещё группы ползователей G Каждый пользователь в какой-либо группе.
Каждый пользователь по-разному влияет на свою группу.
Задача - построить рекомендации для групп пользователей, зная матрицы взаимодействия user-items, groups-items

SLAGR

Оценка товара t для группы l складывается из влияний каждого пользователя (со своим весом) и биаса. H_v, H_u - эмбединги товара и пользователя attention сети

$$z(t,j) = A^T \text{ReLU}(H_v v_t + H_u u_j + b')$$

$$\alpha(t,j) = Softmax(z(t,j))$$

$$g_l(t) = \sum_j \alpha(t,j) u_j + g_l$$

Конкатенируем эмбединги групп и товаров, прогоняем через полносвязные слои и получаем предсказание

$$e_0 = \phi_{pooling}(g_l(t), v_t) = (g_l(t) \odot v_t, g_l(t), v_t)^T$$

$$e_n = \text{Dense}(e_0)$$

$$\hat{(}y_i t) = w^T e_n$$

$$\mathcal{L}(y_{ui} - \hat{y_{ui}}) = \sum_{u,i} (r_{ui} - \hat{r_{ui}})^2$$

SLAGR

Преимущества :

1. Получаем эмбединги для всего
2. Используем социальную информацию и социальную модель
3. При новых взаимодействиях достаточно доучиваться

SLAGR

Преимущества :

1. Получаем эмбединги для всего
2. Используем социальную информацию и социальную модель
3. При новых взаимодействиях достаточно доучиваться

Недостатки:

1. Очень долго учиться
2. Применима ли на практике?