

## Рекомендательные системы

$U = \{u_1, \dots, u_n\}$  - users

$I = \{i_1, \dots, i_m\}$  - items (товары, видео, статьи, ...)

для некоторых  $(u, i) \in U \times I$   $\exists r_{ui}$  - рейтинг  
(результат взаимодействия  $u$  с  $i$ )

- $\{0, 1\}$
- $\mathbb{R}$
- $\{1, \dots, K\}$

Обратно, что считать за рейтинг - отдельная задача

$R = \{(u, i) \mid \exists r_{ui}\}$  - набор факт. взаимодействий

$|R| \ll n \cdot m$

Задача: построить  $a(u, i) \approx r_{ui} \quad \forall u, i$

### 1) Архитектура рек. системы

$u$  запрос

отбор  
кандидатов

$I \rightarrow I(u)$

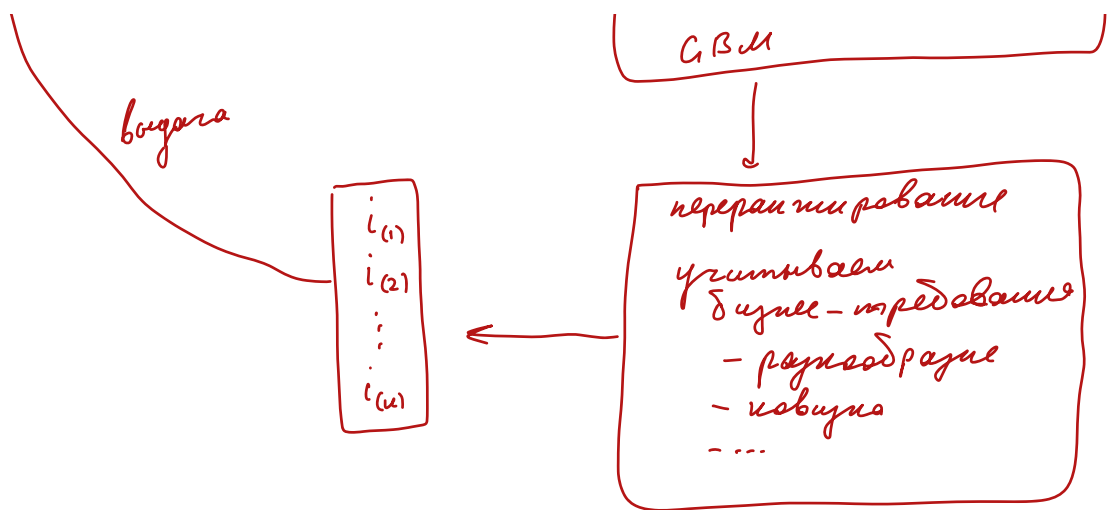
$|I(u)| \approx 10^3 - 10^4$

ранжирование

$(u, i) \rightarrow \underbrace{f(u, i) = x_{ui}}_{\text{признаки}}$

$i \in I(u)$

$\theta(f(u, i)) \approx r_{ui}$



Методы можно использовать для отбора кандидатов или как признаки в ранжировании (в  $f(u, i)$ )

## 2) Коллаборативная фильтрация

используем только информацию о взаимодействиях



### 2.1 Memory-based models user-based

$$I_{uv} = \{i \in I \mid \exists r_{ui} \& \exists r_{vi}\}$$

$$u, v \in U$$

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

исходно  
даны пары.

$$u_0 \rightarrow \mathcal{U}(u_0) = \{v \mid \text{sim}(u_0, v) \geq \alpha\} \quad \text{— user param.}$$

кампания

$$\downarrow$$

$$\text{Score}(i) = \frac{1}{|\mathcal{U}(u_0)|} \sum_{\substack{u \in \mathcal{U}(u_0) \\ \exists r_{ui}}} r_{ui}$$

$\downarrow$   
ранжировка по  $\text{Score}(i)$

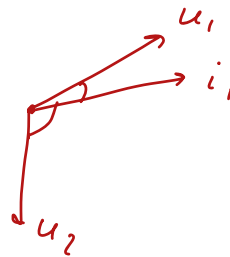
- + простой подход
- проблемы с объёмом памяти
- долго
- не обобщается
- рекоменд. могут оказаться не релевантными

## 2.2 Модели на основе матричных разложений

$$u \rightarrow p_u \in \mathbb{R}^d$$

$$i \rightarrow q_i \in \mathbb{R}^d$$

$$\langle p_u, q_i \rangle \approx r_{ui}$$



$$\sum_{(u,i) \in R} (r_{ui} - \langle p_u, q_i \rangle)^2 \rightarrow \min_{p_u, q_i} \quad (*)$$

log-loss и другая ф-ция потерь

---

как это связано с матричными разложениями?

$$R: R_{ui} = \begin{cases} r_{ui}, & \exists r_{ui} \\ 0, & \nexists r_{ui} \end{cases}$$

|  
матрица

$$P = (p_1 | p_2 | \dots | p_n) \in \mathbb{R}^{d \times n}$$

$$Q = (q_1 | q_2 | \dots | q_m) \in \mathbb{R}^{d \times m}$$

$$r_{ui} \approx \langle p_u, q_i \rangle$$

$\swarrow \quad \searrow$   
 $u\text{-й столбец } P \quad i\text{-й столбец } Q$

$$R \approx P^T Q$$

$n \times m \quad n \times d \quad d \times m$   
 $d \approx 100$

- задача о минимизации  
приближения матрицы

$P$  и  $Q$  можно найти через сингулярное разложение  $R$  (SVD)

поиск - Pure SVD

### 2.3 Неявные модели

проблема:  $p_u$  и  $q_i$  могут много отсутствовать из-за маленького кол-ва рейтингов

решение: если мы не знаем  $r_{ui}$ , то  $r_{ui} := 0$  и поставим ему маленький вес (разрешим модели оптимизироваться)

$$S_{ui} = \begin{cases} r_{ui}, & \exists r_{ui} \\ 0, & \nexists r_{ui} \end{cases}$$

$$C_{ui} = 1 + \alpha \cdot S_{ui} - \text{bias} \rightarrow \begin{matrix} 1 \text{ для "искусств. рейти."} \\ \rightarrow \text{дельный для наст. рейти.} \end{matrix}$$

$\alpha = \frac{50}{100}$

$$\sum_{\substack{u \in U \\ i \in I}} C_{ui} (S_{ui} - \langle p_u, q_i \rangle)^2 \rightarrow \min_{p_u, q_i}$$

③ Учёт контекста

один из вариантов:

$i \xrightarrow{\text{нейросеть}} \text{задан. } v_i \rightarrow \text{добавляет к } f(u, i)$

④ Как это всё делать

④.1 Отбор кандидатов

- эвристич. (подписки, популярное, ...)
- memory-based
- матриц разном., где строка  $I$  - группы (авторы, категории)
- ...

④.2 Ранжирование

$(u, i) \rightarrow \text{признаки} \rightarrow \text{GBM}$

- про  $u$
- про  $i$

- про  $(u, i)$   
| кантун,  
| бокс  
| мод  
| мод  
| мод

## ⑤ Холодный старт (cold start)

что делать с новым **айтемом**?

- exploration / показываем подтестам или показываем на них)
- восстанов. контента

что делать с новым **пользователем**?

- онбординг
- популяризе
- показывать персонализацию подбора