# 协同过滤 -- 集体的智慧

## 也是一种显式反馈的推荐方法

基于内存的协同过滤方法可以分为两个主要部分：User-based CF、Item-based CF

User-based CF 和 Item-based CF都有的
三个重要的问题：
1. 如何计算两个用户或者物品的相似度？　(Similarity Measurement problem?)
2. How to select some similar users or items?　(Neighborhood Selection?)
3. how to predict the rating based on the information of similar users of items? (Prediciton Rule?)

## User-based CF

1. 答：

如何通俗易懂地解释「协方差」与「相关系数」的概念？ - GRAYLAMB的回答 - 知乎
https://www.zhihu.com/question/20852004/answer/134902061

**用户相似度问题：**
**使用皮尔逊相关系数（或者直接说相关系数）来衡量二者的相似度**

Pearson correlation coefficient (PCC) between user $u$ and user $w$,

$$s_{wu} = \frac{\sum_{k \in \mathcal{I}_w \cap \mathcal{I}_u} (r_{uk} - \bar{r}_u)(r_{wk} - \bar{r}_w)}{\sqrt{\sum_{k \in \mathcal{I}_w \cap \mathcal{I}_u} (r_{uk} - \bar{r}_u)^2} \sqrt{\sum_{k \in \mathcal{I}_w \cap \mathcal{I}_u} (r_{wk} - \bar{r}_w)^2}} \quad (1)$$

Notes:

- $-1 \leq s_{wu} \leq 1$

相关系数=样本的协方差/两标准差之积（分子分母都约去了1/n，n是Iw和Iu两集合的交集的元素个数）

Swu 越靠近1，两者正相关度越大， -1则负相关(相反变化)，0则无相关

2. 答:

找到 Top K 个最近邻居

- Similarity threshold
- Top-$K$ most nearest neighbors
  - Step 1. Obtain the neighbors of user $u$ where $s_{wu} \neq 0$, i.e., $\mathcal{N}_u$
    - In practice, we usually use a large $\mathcal{N}_u$ as candidate users (**instead of all the neighbors**) due to the high space cost
  - Step 2. Obtain the users who rated item $j$, i.e., $\mathcal{U}_j$
  - Step 3. Obtain a set of top-$K$ nearest neighbors of user $u$ from $\mathcal{U}_j \cap \mathcal{N}_u$ (when estimating the rating of $\hat{r}_{uj}$), i.e., $\mathcal{N}_u^j \subseteq \mathcal{U}_j \cap \mathcal{N}_u$ with $|\mathcal{N}_u^j| = K$

3. 答:

Predicted rating of user $u$ on item $j$,

$$\hat{r}_{uj} = \bar{r}_u + \frac{\sum_{w \in \mathcal{N}_u^j} s_{wu}(r_{wj} - \bar{r}_w)}{\sum_{w \in \mathcal{N}_u^j} s_{wu}} \qquad (2)$$

Notes:

- sometimes, we will use the following prediction rule,

$$\hat{r}_{uj} = \bar{r}_u + \frac{\sum_{w \in \mathcal{N}_u^j} s_{wu}(r_{wj} - \bar{r}_w)}{\sum_{w \in \mathcal{N}_u^j} |s_{wu}|}$$

- the default value is $\bar{r}_u$ if $\mathcal{N}_u^j = \emptyset$
- $\mathcal{N}_u^j$ is dependent on both user $u$ and item $j$

## Item-based CF

1. 答:

**物品相似度问题:**

**使用 调整过后的 余弦相似度 公式计算**

Adjusted Cosine similarity between item $k$ and item $j$,

$$s_{kj} = \frac{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} (r_{uk} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} (r_{uk} - \bar{r}_u)^2} \sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} (r_{uj} - \bar{r}_u)^2}} \tag{3}$$

Notes

- $-1 \leq s_{kj} \leq 1$
- Cosine similarity between item $k$ and item $j$

$$s_{kj} = \frac{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} r_{uk} r_{uj}}{\sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} r_{uk}^2} \sqrt{\sum_{u \in \mathcal{U}_k \cap \mathcal{U}_j} r_{uj}^2}}$$ 原本的余弦相似度公式

2. 答:

- Similarity threshold

- Top-$K$ most nearest neighbors
  - Step 1. Obtain the neighbors of item $j$ where $s_{kj} \neq 0$, i.e., $\mathcal{N}_j$
    - In practice, we usually use a large $\mathcal{N}_j$ as candidate items (**instead of all the neighbors**) due to the high space cost
  - Step 2. Obtain the items rated by user $u$, i.e., $\mathcal{I}_u$
  - Step 3. Obtain a set of top-$K$ nearest neighbors of item $j$ from $\mathcal{I}_u \cap \mathcal{N}_j$ (when estimating the rating of $\hat{r}_{uj}$), i.e., $\mathcal{N}_j^u \subseteq \mathcal{I}_u \cap \mathcal{N}_j$ with $|\mathcal{N}_j^u| = K$
  - $K$ is a parameter needs to be tuned, e.g., $K \in \{20, 30, 40, 50, 100\}$

3. 答:

Predicted rating of user $u$ on item $j$,

$$\hat{r}_{uj} = \frac{\sum_{k \in \mathcal{N}_j^u} s_{kj} r_{uk}}{\sum_{k \in \mathcal{N}_j^u} s_{kj}} \tag{4}$$

Notes:

- the default value is $\bar{r}_u$ if $\mathcal{N}_j^u = \emptyset$
- $\mathcal{N}_j^u$ is dependent on both item $j$ and user $u$

## Hybrid_CF:

Predicted rating of user $u$ on item $j$,

$$\hat{r}_{uj} = \lambda^{UCF} \hat{r}_{uj}^{UCF} + (1 - \lambda^{UCF}) \hat{r}_{uj}^{ICF}$$

where $0 \leq \lambda^{UCF} \leq 1$ is a tradeoff parameter.

**同样 也需要 衡量误差 借助 测试集**

- Mean Absolute Error (MAE)

$$MAE = \sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} |r_{ui} - \hat{r}_{ui}| / |\mathcal{R}^{te}|$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\sum_{(u,i,r_{ui}) \in \mathcal{R}^{te}} (r_{ui} - \hat{r}_{ui})^2 / |\mathcal{R}^{te}|}$$

- Performance: the smaller the better.

个人实现结果：　可以通过（0.4%偏差）

```
User-based CF:
RMSE:  0.9599   MAE:  0.7516
Item-based CF:
RMSE:  0.9883   MAE:  0.7789
Hybrid-based CF:
RMSE:  0.961   MAE:  0.757
```

Slide 结果：

| Method | RMSE | MAE |
|---|---|---|
| User-based CF | **0.9554** | **0.7480** |
| Item-based CF | 0.9901 | 0.7801 |
| Hybrid CF | 0.9562 | 0.7538 |

Observation: Hybrid CF and user-based CF perform better than item-based CF on this data.