数据挖掘干货总结（六）--推荐算法之CF

原创 Shuan Xi 大数据干货分享 2018-05-01

本文共计1245字，预计阅读时长八分钟

推荐算法(二）--CF算法

一、推荐的本质

推荐分为非个性化和个性化，非个性化推荐比如各类榜单，而本系列主要介绍**个性化推荐**，即：

在合适的场景，合适的时机，通过合适的渠道，把合适的内容，推荐给合适的用户

二、推荐算法的种类

1. 基于内容Content Based
2. 基于协同Collaboration Filtering

– User Based CF

– Item Based CF

三、CF算法详解

1. 原理框架

原理：

① User-Based CF

a.假设用户喜欢那些跟他有相似爱好的用户喜欢的东西

b.假设具有相似兴趣的用户在未来也具有相似兴趣

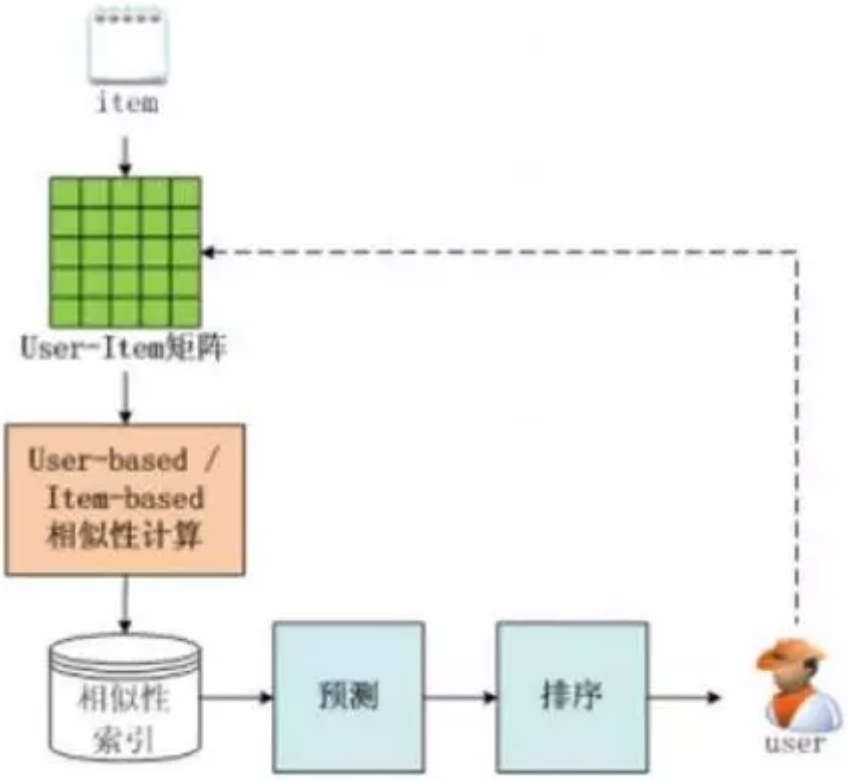
c.给定用户u，找到一个用户的集合N(u)，他们和u具有相似的兴趣，将N(u)喜欢的物品推荐给用户

②Item-Based CF

a.假设用户喜欢跟他过去喜欢的物品相似的物品

b.假设历史上相似的物品在未来也相似

c.给定用户u，找到他过去喜欢的物品的集合R(u)，把和R(u)相似的物品推荐给u.



优点：

a.充分利用群体智慧

b.推荐精度高于CB

c.利用挖掘的隐含相关性

缺点：

a.解释性较差

b.对时效性强的item不适用

c.冷启动问题

2. 处理过程：

① 数据准备：

用户user_id，物品item_id，打分score（score可以是用户对某件物品的评分，或者是根据用户行为计算出的偏好度得分，如曝光，点击，收藏的加权得分，具体权重可以参考漏斗模型），如下：

user_id item_id score

id1 item1 3

id1 item2 2

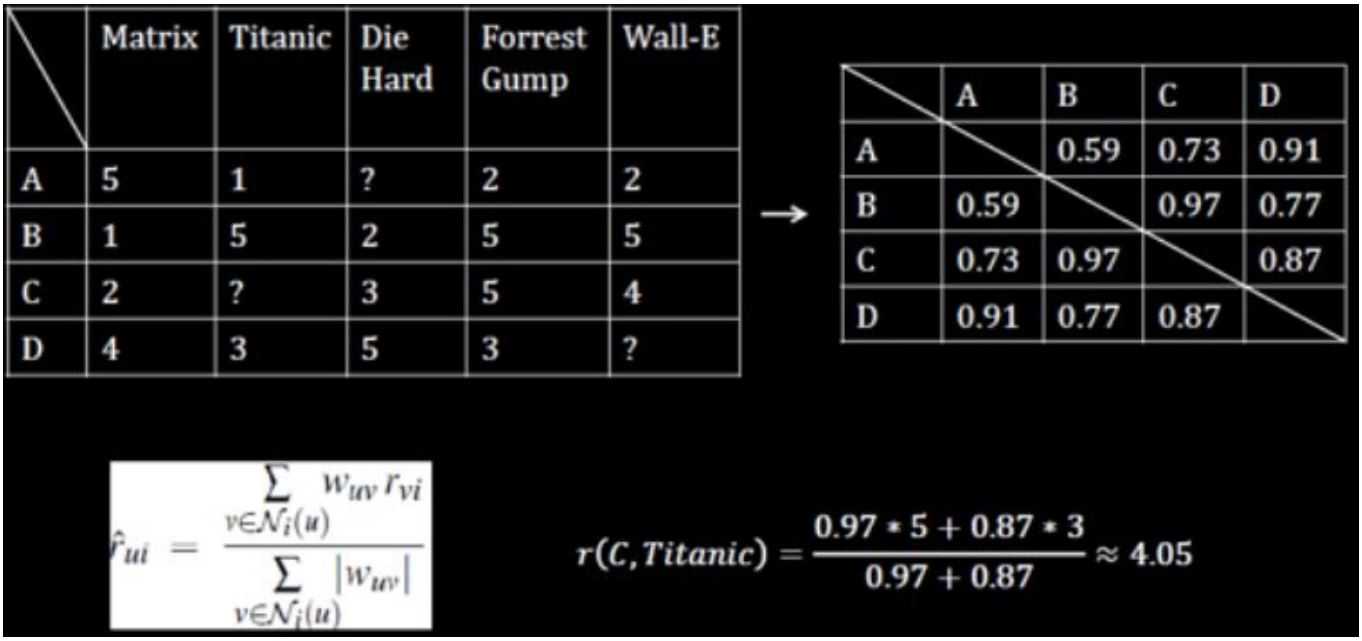
id2 item2 2

id3 item3 4

... ...

②计算相似性矩阵：

CF算法的关键在于得到user或item的相似度矩阵，下面以User_Based为例。



用户之间的相似度计算，是基于对相同的物品打过分，可以将各个分值，联合起来作为一个向量，然后计算余弦相似度：

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

去重后即得到上面的相似度矩阵。

③推荐：

根据相似度矩阵，选择与目标用户相似度最高的几位用户，在第一张表中选取各自打分较高的物品，形成一个推荐候选集合，准备推荐给目标用户（对于新闻，电影之类的数据可以在去重后再推送）

注：如果是Item_Based的算法，则是计算各个item之间的相似度矩阵，即对两个item都打过分的id的打分情况作为向量，同理得到item的相似度矩阵。

item1 [id1:2, id2:3, id5:2] (2,3,2)

item2 [id1:3, id2:1, id5:1] (2,1,1)

3. 问题：

① 冷启动及对应方案

case1：用户冷启动

- 提供**热门排行榜**，等用户数据收集到一定程度再切换到个性化推荐
- 利用用户注册时提供的年龄、性别、IP、登录时间等数据做**粗粒度的个性化**
- 利用用户社交网络账号，导入用户在**社交网站上的好友信息**，然后给用户推荐其好友喜欢的物品
- 在用户新登录时要求其对一些物品进行反馈，**收集这些兴趣信息**，然后给用户推荐相似的物品

case2：物品冷启动

• 将新物品推荐给可能对它感兴趣的用户，利用内容信息，将他们推荐给喜欢过和它们相似的物品的用户
• 物品必须能够在第一时间展现给用户，否则经过一段事件后，物品的价值就大大降低了
• UserCF和ItemCF都行不通，只能利用Content based解决该问题，频繁更新相关性数据

case3：系统冷启动

• 引入专家知识，通过一定高效方式迅速建立起物品的相关性矩阵

② Item_Based和Content-Based的区别

• 区别在于的CB中相似度是根据**item**的属性向量计算得到，而CF中是根据所有用户对**item**的评分向量计算得到。

③ ALS 交替最小二乘（Alternating Least Squares）

• 算是ml中对CF算法的一种优化
• 对于一个users-items-score的评分数据集，ALS会建立一个users*items的m*n的矩阵。其中，m为users的数量，n为items的数量。
• 这个数据集中，并不是每个用户都对每个产品进行过评分，所以这个矩阵往往是稀疏的，用户i对产品j的评分往往是空的。
• ALS所做的事情就是将这个稀疏矩阵通过一定的规律填满，这样就可以从矩阵中得到任意一个user对任意一个item的评分，ALS填充的评分项也称为用户i对物品j的预测得分。

以上。

ps：
CF代码数量较多
这次就没附上
需要的同学私信我就好

pps：
数据挖掘系列的笔记都整理好了
包括往期的NLP，分类，聚类算法
有需要的同学可以去菜单看看

每天向你推送最实用的干货，记得收藏点赞置顶哦~