电子商务网站是个性化推荐系统重要地应用的领域之一，亚马逊就是个性化推荐系统的积极应用者和推广者，亚马逊的推荐系统深入到网站的各类商品，为亚马逊带来了至少30%的销售额。

不光是电商类，推荐系统无处不在。QQ，人人网的好友推荐；新浪微博的你可能感觉兴趣的人；优酷，土豆的电影推荐；豆瓣的图书推荐；大从点评的餐饮推荐；世纪佳缘的相亲推荐；天际网的职业推荐等。

**推荐算法分类：**

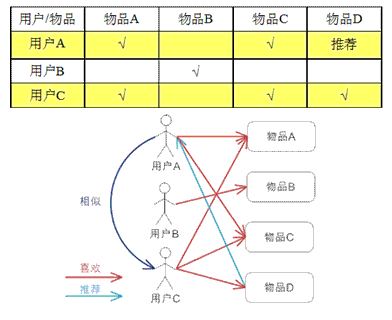
按数据使用划分：

* 协同过滤算法：UserCF, ItemCF, ModelCF
* 基于内容的推荐: 用户内容属性和物品内容属性
* 社会化过滤：基于用户的社会网络关系

**基于用户的协同过滤算法UserCF**

基于用户的协同过滤，通过不同用户对物品的评分来评测用户之间的相似性，基于用户之间的相似性做出推荐。简单来讲就是：给用户推荐和他兴趣相似的其他用户喜欢的物品。

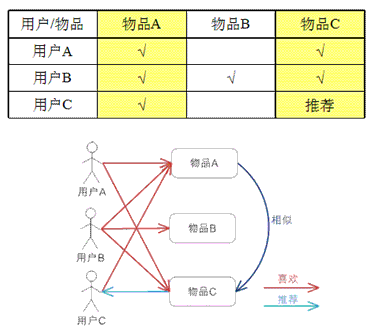
用例说明：

[](http://blog.fens.me/wp-content/uploads/2013/10/image015.gif)

**基于物品的协同过滤算法ItemCF**

基于item的协同过滤，通过用户对不同item的评分来评测item之间的相似性，基于item之间的相似性做出推荐。简单来讲就是：给用户推荐和他之前喜欢的物品相似的物品。

用例说明：

[](http://blog.fens.me/wp-content/uploads/2013/10/image017.gif)

如用户c买过物品a,没买过物品b、c,那么看物品a和物品b的重合度,物品a和物品c的重合度,然后分别乘以用户c对物品b的喜爱度和对物品c的喜爱度,最后分值高的就是和物品a最相似的,推荐给用户c

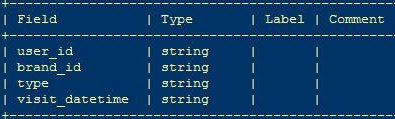
注：基于物品的协同过滤算法，是目前商用最广泛的推荐算法。

案例：天池大数据竞赛

**我们会开放如下数据类型：**

|  |  |  |
| --- | --- | --- |
| **字 段** | **字段说明** | **提取说明** |
| **user\_id** | **用户标记** | **抽样&字段加密** |
| **Time** | **行为时间** | **精度到天级别&隐藏年份** |
| **action\_type** | **用户对品牌的行为类型** | **包括点击、购买、加入购物车、收藏4种行为  (点击：0 购买：1 收藏：2 购物车：3）** |
| **brand\_id** | **品牌数字ID** | **抽样&字段加密** |

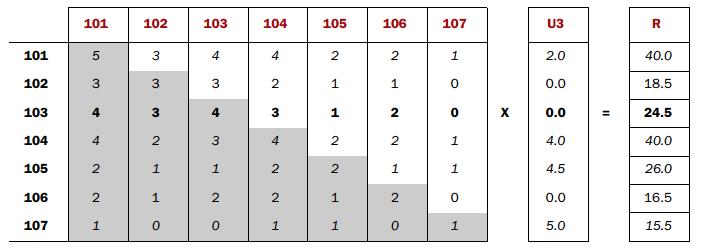
     提供的数据量，涉及千万级天猫用户，万级天猫品牌，时间跨度4个月的行为记录。   
     提供的训练数据在天池集群的表t\_alibaba\_bigdata\_user\_brand\_total\_1中，字段分别为：user\_id，brand\_id, type, visit\_datetime。如图所示



     用户4种行为类型(Type)对应代码分别为：   
     点击：0；购买：1；收藏：2；购物车：3

数据案例为：

再谈谈Co-occurrence Matrix(同显矩阵)和User Preference Vector(用户评分向量)相乘得到的这个**R**ecommended Vector(推荐向量)的意义



第一次听完理论直接上这两个东西相乘实现Item Based Cooperative Filtering(基于物品的协同过滤)，一下子真没反应过来，下面就个人的理解通俗的解释一下：

ItemBased：基于物品的（区分于基于用户的）体现在同现矩阵，把所有用户对物品打过分的记录都拿过来，形成一个个反应物品关联度的矩阵Co-occurrence Matrix，下面简称C矩阵。

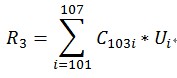
为什么乘以User Preference Vector用户评分向量就是**R**ecommended Vector(推荐向量)，这个推荐向量又要怎么用呢？

还是用R的第三项24.5来做一下解释，

R3的解释：对于用户U商品103的可推荐度。

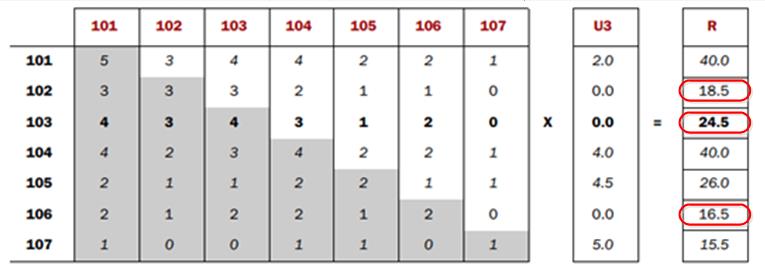
这点很重要，理解这点就是要知道我们这一系列算法过程在做什么（What）。

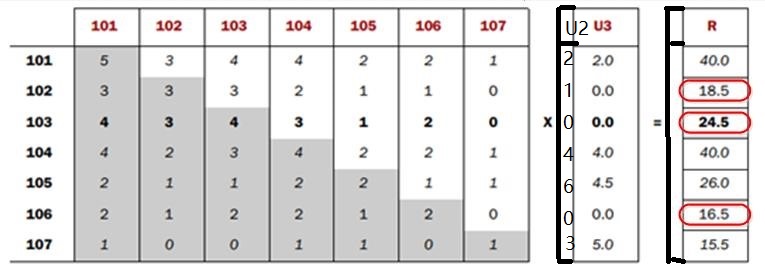
我把R3也就是R103的计算用公式表示如下：



R3怎么出来的

从上面可以看到C103i\*Ui就是Ui代表用户对i的喜爱度，C103i代表i和103同时出现的次数，i物品和103同时出现得越多C103i越大，用户对i的喜爱度值越大Ui越大，自然R103值就越大，越值得推荐103。





R向量里面的R101, R104,R105和R107这三项值很大，但是我们可以忽略它们应用用户已经对该物品购买过的，也就是已经买个这些了，可以不推荐了，对于用户没有买过的几项里面选出最大(或者TopN)的物品推荐就可以了，

上面R102,R103,R106里面选一个最大值103,103就是可以推荐的商品了

上面这个图的意思解释:

U3从上往下是对101~107的喜爱度,横坐标是103的那一行和U3相乘得到的值是物品103对用户U3的推荐度,上面的图只列出U3,其实还有很多其他用户,所以其实是两个矩阵相乘

红色的101~107就相当于第四步MR中的map输出key

黑色的101~107就是要算的对用户推荐度是多少的商品

代码步骤：

1. 去重(原始数据中有重复的数据(两行数据4个列值都一样))
2. 获取所有用户对所有物品的喜欢矩阵：

|  |  |  |  |
| --- | --- | --- | --- |
|  | I1 | I2 | I3 |
| U1 |  |  |  |
| U2 |  |  |  |

公司里是用逻辑回归算法算出用户对物品的喜爱度,这里先不考虑逻辑回归,认为点击喜爱度为1,收藏为2,加入购物车为3,购买为4.

1. 获得所有物品之间的同现矩阵
2. 两个矩阵相乘得到三维矩阵
3. 三维矩阵的数据相加获得所有用户对所有物品的推荐值（二维矩阵）
4. 按照推荐值降序排序。