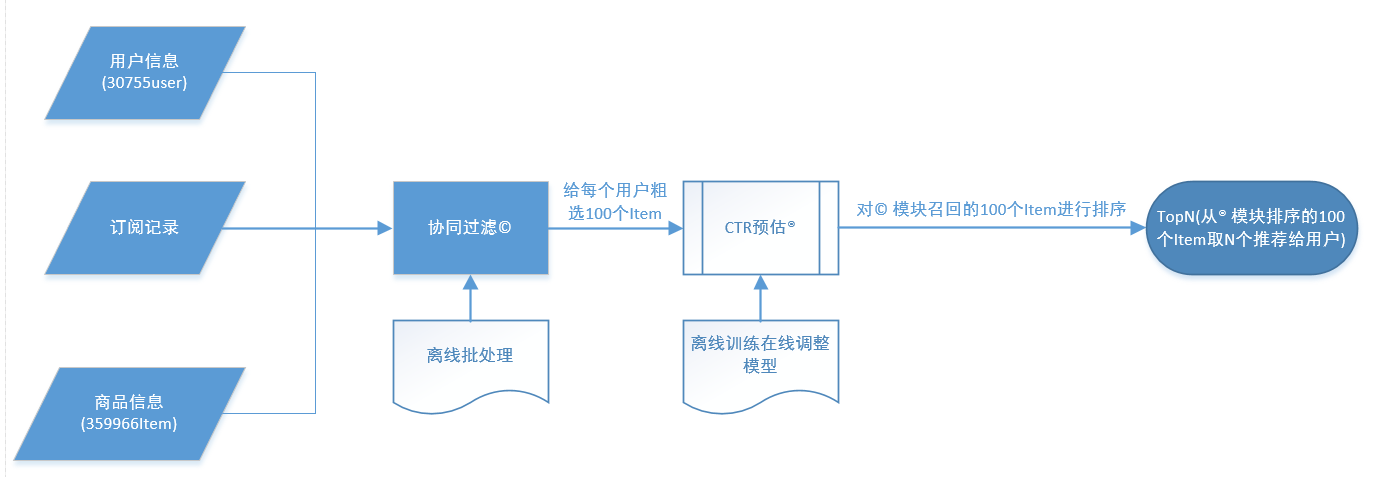
推荐系统最主要有召回和排序2大环节，召回可以理解为向用户粗选一批待推荐的商品，相当于粗排序，之后会加一层CTR预估的rank模型，相当于精排序，这样切分的主要目的在于加快推荐速度：召回模块从所有Item中海选一批给排序环节做CTR预估取TopN，rank模型只对召回阶段输出的少量Item(对最终推荐给用户的数量来说依然很多)进行排序，避免了对所有Item的全排序，从而加快了推荐效率。

召回一般有基于内容，基于关联规则和基于协同过滤等手段，初步筛选出用户有可能感兴趣的Item。

协同过滤算法具体实现的时候, 又分为典型的两类:

1. 基于领域的协同过滤算法:这类算法的主要思想是利用<user,item,score>打分矩阵, 利用统计信息计算用户和用户, item和item之间的相似度。然后再利用相似度排序, 最终得出推荐结果。主要有基于Item和基于User相似度2种协同。
2. 基于模型的协同过滤算法，有基于矩阵分解和潜在语义，贝叶斯网络以及SVM等。

本系统拟采用系统过滤进行召回，主要流程如下：



原始数据如下：

|  |  |  |
| --- | --- | --- |
| 用户 | 音乐 | 是否续订 |
| user1 | item1 | 0 |
| user1 | item3 | 1 |
| user2 | item2 | 0 |
| user2 | item4 | 1 |
| user3 | item1 | 1 |
| user3 | item3 | 0 |
| user3 | item5 | 0 |
| user4 | item1 | 0 |
| user5 | item2 | 1 |
| user5 | item3 | 0 |
| user5 | item4 | 0 |
| user5 | item5 | 1 |

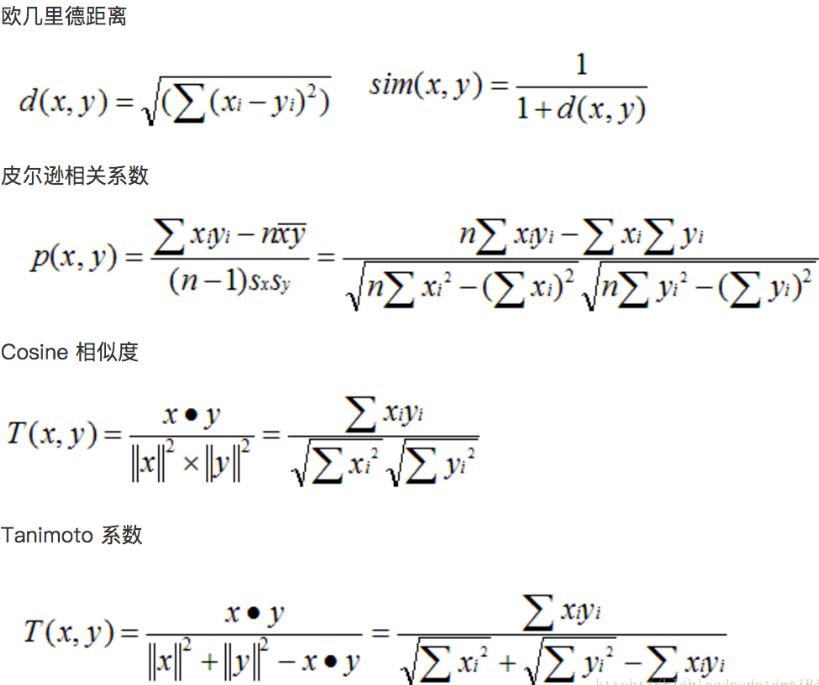
表1.1 训练样本

将表1.1数据转换成user-item评分矩阵，如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | item1 | item2 | item3 | item4 | item5 |
| user1 | 0 |  | 1 |  |  |
| user2 |  | 0 |  | 1 |  |
| user3 | 1 |  | 0 |  | 0 |
| user4 | 0 |  |  |  |  |
| user5 |  | 1 | 0 | 0 | 1 |

表1.2 user-item打分表:其中1表示续订（喜欢），0不续订（不喜欢），空表示无关联

根据各用户对item的打分，计算item之间的相似度。由表1.2可见，各用户打分只有0和1以及未浏览共3种无量纲分值，用欧氏距离，皮尔逊相关系数，余弦相似度，Tanimoto系数（广义Jaccard相似系数）计算相似度均可以。



欧氏距离相似度：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | user1 | user2 | user3 | user4 | user5 |
| user1 | 1 | 0 |  | 1 | 0.5 |
| user2 |  | 1 | 0 | 0 |  |
| user3 |  |  | 1 | 0.5 | 0.5 |
| user4 |  |  |  | 1 | 0 |
| user5 |  |  |  |  | 1 |

表1.3 用户欧氏距离相似度

Cosine相似度：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | user1 | user2 | user3 | user4 | user5 |
| user1 | 1 | 0 | 0 | NaN | 0 |
| user2 |  | 1 | 0 | NaN | 0 |
| user3 |  |  | 1 | NaN | 0 |
| user4 |  |  |  | NaN | NaN |
| user5 |  |  |  |  |  |

表1.4 用户Cosin相似度

Jaccard相似系数：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | user1 | user2 | user3 | user4 | user5 |
| user1 | 1 | 0 | 0 | 0 | 0 |
| user2 |  | 1 | 0 | 0 | 0 |
| user3 |  |  | 1 | 0 | 0 |
| user4 |  |  |  | 0 | 0 |
| user5 |  |  |  |  |  |

表1.5 用户Jaccard相似系数

由于user-item矩阵特别稀疏，会出现很多不合理的相似度值，如表1.3中红色部分，user1和user4由于稀疏导致相似度100%，实际上他们仅有item1的反馈是一样，由于其他item各有未打分，因而无法确认他们分值是否一致，所以他们100%相似是不合理的。总之，由于评分矩阵特别稀疏，导致大量评分用不上，如此处的user4与user5的相似度计算中，user5对item2,item3,item4,item5的评分根本没有用上。此外相似度还缺乏统一的评价指标，不同的计算公式得到的相似度是不一样的，如表1.3，与user1最相似的是user5,而表1.4，表1.5中很多user的相似度由于评分稀疏而无法计算。可见手工计算相似度相当困难，即使计算出来，结果也可能不可信，所以本系统通过训练模型来学习item之间的相似度。

由原始训练数据表1.1整理出各user已续订了的item（target=1），用户续订表如下：

|  |  |  |
| --- | --- | --- |
| 用户已续订 | | |
| user1 | item3 | |
| user2 | item4 | |
| user3 | item1 | |
| user4 |  | |
| user5 | item2 | item5 |

表1.6 用户续订表

我们为了计算item与item之间的关系，可以结合表1.6的用户续订信息，扩展表1.1的原始数据，使其只包含item与item的交叉特征来拟合target，从而训练出一个二分类模型，根据此模型各交叉特征的权重系数，将得到item×item的关系系数，作为item之间的相似度。



表1.7 训练item相似度的训练集

表1.1和表1.6扩展为表1.7的映射规则如下：

第1行样本，user1未续订item1(根据表1.1)且已续订了item3(根据表1.6)，所以item1×item3的特征值为1，其他特征值为0,输出分类标签为0；

第2行样本，user1续订了item3(根据表1.1)且已续订了item3(根据表1.6)，所以item3×item3的特征值为1，其他特征值为0,输出分类标签为1；

第3行样本，user2未续订item2(根据表1.1)且已续订了item4(根据表1.6)，所以item2×item4的特征值为1，其他特征值为0,输出分类标签为0；

第4行样本，user2续订了item4(根据表1.1)且已续订了item4(根据表1.6)，所以item4×item4的特征值为1，其他特征值为0,输出分类标签为1；

第5行样本，user3续订了item1(根据表1.1)且已续订了item1(根据表1.6)，所以item1×item1的特征值为1，其他特征值为0,输出分类标签为1；

第6行样本，user3未续订item3(根据表1.1)且已续订了item1(根据表1.6)，所以item1×item3的特征值为1，其他特征值为0,输出分类标签为0；

第7行样本，user3未续订item5(根据表1.1)且已续订了item1(根据表1.6)，所以item1×item5的特征值为1，其他特征值为0,输出分类标签为0；

第8行样本，user4未续订item1(根据表1.1)且未续订任何item，所以所有交叉特征值均为0,输出分类标签为0；

第9行样本，user5续订了item2(根据表1.1)且已续订了item2和item5(根据表1.6)，所以item2×item2和item2×item5的特征值为1，其他特征值为0,输出分类标签为1；

第10行样本，user5未续订item3(根据表1.1)且已续订了item2和item5(根据表1.6)，所以item2×item3和item3×item5的特征值为1，其他特征值为0,输出分类标签为0；

第11行样本，user5未续订item4(根据表1.1)且已续订了item2和item5(根据表1.6)，所以item2×item4和item4×item5的特征值为1，其他特征值为0,输出分类标签为0；

第13行样本，user5续订了item5(根据表1.1)且已续订了item2和item5(根据表1.6)，所以item2×item5和item5×item5的特征值为1，其他特征值为0,输出分类标签为1；

…

总体规则就说已续订标签为1未续订标签未0，至于交叉特征是0还是1，则根据特征中是否存在已续订的item。

根据以上数据集训练得到一个二分类模型，模型特征权重即为item的相似度，权重越大，相似度就越大。将权重稀疏填入如下对称矩阵：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | item1 | item2 | item3 | item4 | item5 |
| item1 | W11 | W12 | W13 | W14 | W15 |
| item2 | W12 | W22 | W23 | W24 | W25 |
| item3 | W13 | W23 | W33 | W34 | W35 |
| item4 | W14 | W24 | W34 | W44 | W45 |
| item5 | W15 | W25 | W35 | W45 | W55 |

表1.8 权重系数即item相似度

根据表1.8进行Item-Based CF召回

如用户userx对item1,item2和item4的评分反馈分别为s1,s2,s3，那给userx推荐item3的强烈程度为score3=W13\* s1+ W23\* s2+ W34\*s4, 推荐item5的强烈程度为 score5= W15\* s1+ W25\* s2+ W45\*s4。