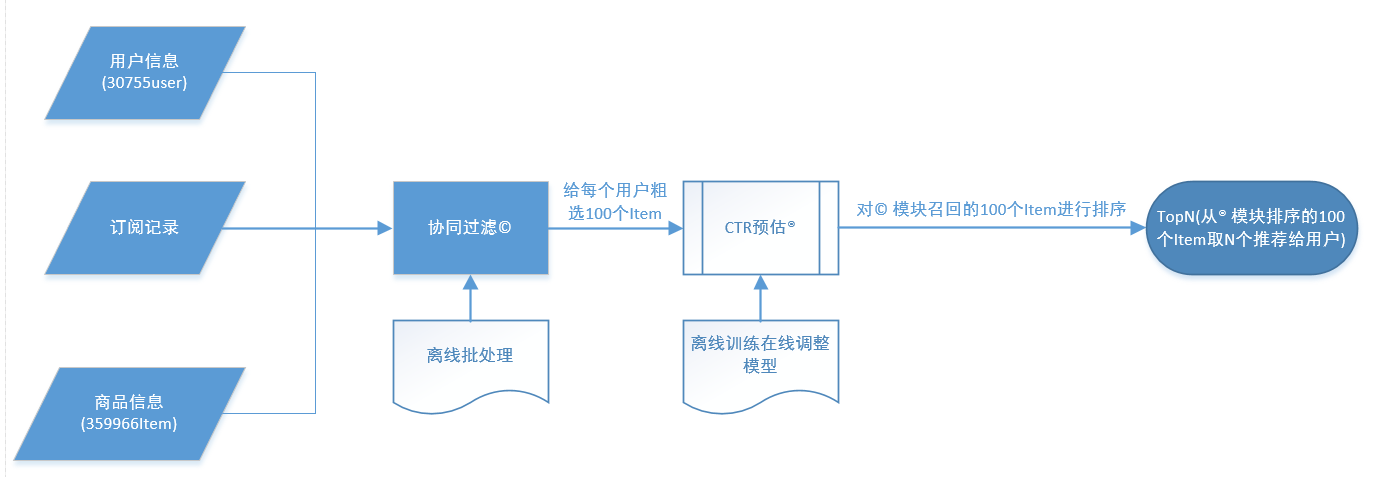
推荐系统最主要有召回和排序2大环节，召回可以理解为向用户粗选一批待推荐的商品，相当于粗排序，之后会加一层CTR预估的rank模型，相当于精排序，这样切分的主要目的在于加快推荐速度：召回模块从所有Item中海选一批推送给排序环节做CTR预估取TopN，rank模型只对召回阶段输出的少量Item(对最终推荐给用户的数量来说依然很多)进行排序，避免了对所有Item的全排序，从而加快了推荐效率，另外召回阶段的item列表还可以人为修改，比如给投放广告费多的item强行插队到召回表中，增加其曝光率，当然也可以在CTR取TopN环节进行最终的推荐列表进行干预。

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, 典型的两类又分为:

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, 根据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，如：基于矩阵分解和隐语义，贝叶斯网络以及SVD等。

本系统采用协同过滤进行召回，主要流程如下：



1. **对数据的理解和分析**
2. 原始数据如下：

msno: 用户id，加密String

song\_id: song id，歌曲id

source\_system\_tab: 触发事件的类型/tab，用于表示app的功能类型

source\_screen\_name: 用户看到的布局的名字（name of the layout）

source\_type: 用户在app上播放音乐的入口的类型

target: 标签。1表示用户在第一次听音乐后会在一个月内继续订阅，0表示没有订阅。

经过数据探索(其他探索详见[ipynb](http://192.168.0.101:8888/notebooks/csdn/Music Recommendation/Untitled Folder/DataExplore.ipynb" \t "http://192.168.0.101:8888/tree/csdn/Music%20Recommendation/_blank))，没有发现用户对音乐的显式评分，只能用是否续订这个隐式反馈作为评分了，从train.csv抽取协同过滤阶段需要的数据如下：

|  |  |  |
| --- | --- | --- |
| 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

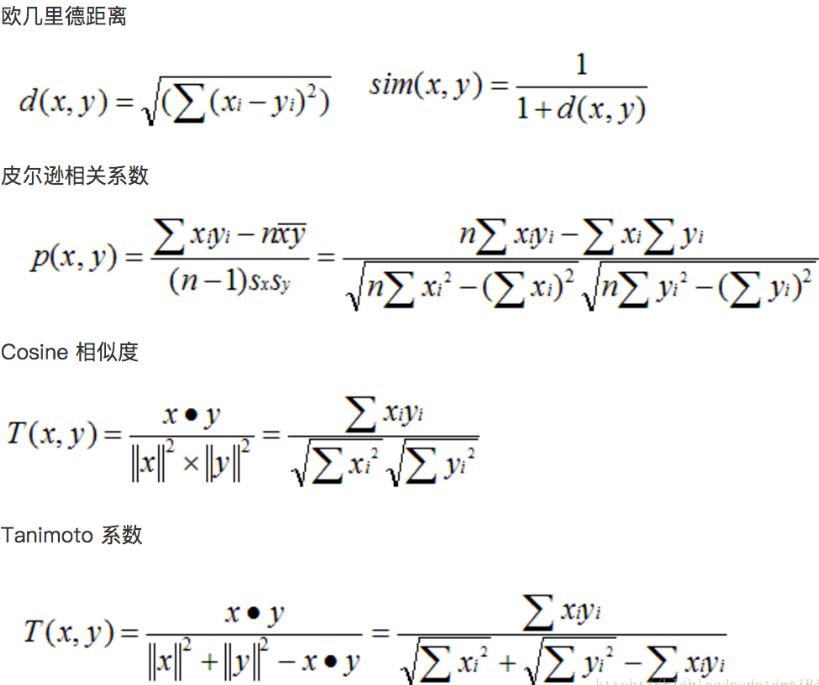
2）将表1.1数据转换成user-item评分矩阵（下称R矩阵），如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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:其中1表示续订（喜欢），0不续订（不喜欢），空表示无关联

**二、相似度计算方法**

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，user对item的打分只有0和1以及从未订阅共3种无量纲值，用欧氏距离，皮尔逊（Pearson）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。



3-1）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

3-2）Pearson相似系数：

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

表1.4 用户Pearson相似系数

3-3）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 |  |  |  | 1 | NaN |
| user5 |  |  |  |  | 1 |

表1.5 用户Cosin相似度

3-4）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 |  |  |  | NaN | 0 |
| user5 |  |  |  |  |  |

表1.6 用户Jaccard相似系数

在数据稀疏的情况，会出现很多不合理的相似度值（已红色标注）：

1. 表1.3中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，而且此处user4只听过一首歌，属于不活跃用户，假如有个活跃用户听过很多歌，而其中恰好有一首与user这首歌相同，那将会出现该用户与user4最相似的错误结论。
2. 各计算方法都有不同的限制：如Pearson对方差为0的评分向量和Cosine对全打0分的用户都无法计算其相似度，由于本系统用户打分只有0分（不续订）和1分（续订）2种情况,所以出现用户全打0分或1分的用户应该挺多的，相似度矩阵将会大量出现如表1.4,表1.5,表1.6所示的NaN值（表示无法计算）。

由于存在以上等问题，导致对“谁跟我相似”缺乏合理标准，所以本系统打算通过训练模型来学习item之间的相似度，避免手工计算相似度，“谁跟我相似”由模型根据历史数据统计出来。

**三、训练模型计算相似度**

根据表1.1整理出各user已续订了的item（target=1），用户续订表如下：



表1.7 用户续订表

从另一个角度看表1.1及其加工出来的表1.7：

user1不续订item1而续订item3；user2不续订item2而续订item4；user3不续订item3，item3而续订item1；user5不续订item3，item4而续订item2，item5...

续不续订都证明用户听过或浏览过这首歌，不续订表示浏览过但不喜欢，续订则表示听过且喜欢，那是否可以像FM一样构造一个两两交叉的变量(交叉特征中存在续订的item则为1，否则为0)？比如用户u听过音乐a,b（但没有续订），续订了音乐c,d,，将得到如下类似的稀疏矩阵：

听了a后：a×c和a×d为1，其他交叉特征为0，标签为0(不喜欢);

听了b后：b×c和b×d为1，其他交叉特征为0，标签为0(不喜欢)；

听了c后：c×c和c×d为1，其他交叉特征为0，标签为1(喜欢)；

听了d后：d×c和d×d为1，其他交叉特征为0，标签为1(喜欢)；

按照这样的映射规则，那表1.1和表1.6扩展为下表：



表1.8 训练item相似度的训练集

训练数据拓展为表1.8所示后，将得到一个二分类问题，训练这个二分类模型，将得到一个item与item两两交叉特征的权重系数，这个权重系数就可以作为item与item之间的相似度系数，将权重稀疏填入得到如下对称矩阵：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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1.9 权重系数即item相似度

根据表1.9进行Item-Based CF召回的规则，就是根据用户以往的续订记录，找出与以往续订item最相似的未续订item，如：

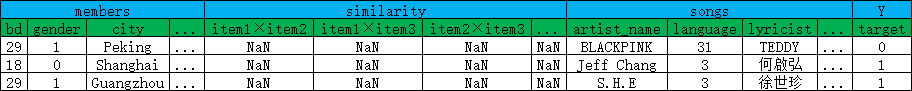
用户user\_i续订了item1,item2和item4，那给user\_i推荐item3的强度为s3=W13+W23+W34(假设系统只有item[1-5]个音乐，将user\_i续订历史记录向量化后变成***u***=[1,1,0,1,0],item3与其他item的相似系数向量后变成***i3***=[W13,W23,W33,W34,W35]，那么s3=***ud·i3***), 同理：推荐item5的强烈程度为s5=***ud·***[W15,W25,W35,W45,W55]=W15+W25+W45，s3和s5谁大则推荐谁,又如用户user\_j续订了item3和item5,那给user\_j推荐item1,item2和item4的相似度分别为：

[0,0,1,0,1]***·***[W11,W12,W13,W14,W15]=W13+W15,

[0,0,1,0,1]***·***[W12,W22,W23,W24,W25]=W23+W25,

[0,0,1,0,1]***·***[W14,W24,W34,W44,W45]=W34+W45。

对于新用户，没有历史记录，则无法用上述协同过滤,此时就必须用其他手段，比如聚类等。如果在之前训练item×item交叉特征权重系数（此处作为item之间的相似度）的二分类模型Model中，额外加入User和Item基本信息，那该模型进行是否续订预测的训练过程中，应该学习到了这些基本信息，于是，即使用户没有续订记录，也仅仅是没有交叉特征而已，但仍然有基本信息等特征可用，Model依然能根据该用户和Item的基本信息，推断出续订概率最大的音乐，如下表：假如有新注册的29岁广州男会员，没有订阅记录，相似度信息无法使用，但根据下表可知推荐徐世珍作词S.H.E演唱的歌曲最有可能被订阅，即该Model还具有一定的泛化能力，此处Model主要任务是训练得到item×item相似度系数，其实也可以作为CTR模型，这样看来召回与排序阶段可以合并？



该模型存在的问题是，为了得到item之间的关联系数（作为相似度），item特征需要两两交叉，本系统至少存在359966首歌，两两交叉将需要359966+359965+359964+……+3+2+1=64,787,940,561个交叉特征，特征膨胀成天文数字，机器性能不允许,放弃此方案。

1. **基于矩阵分解的协同过滤**

对评分矩阵行分解/降维（亦被称为隐因子分解LFM），对比基于用户/物品的协同过滤，没有训练过程，不用计算物品或用户之间的相似度，预测时用分解出来的低维隐因子矩阵简单的相加或相乘，即可近似得到各user对item的评分。总体分解/降维原则就是使逆向还原的矩阵与原R矩阵误差尽可能小，分解矩阵很灵活，可以增加正则项，偏差项和带历史隐式反馈等，求解可以通过梯度下降和采用坐标轴下降法等。



此方法求解出来的R矩阵缺少可解释性，但克服了手动或训练交叉特征求相似度等方案存在的问题，而且很多开源库可用（如Spark的ALS），不用重复造轮子，所以本系统最终采用此方案。