系统功能模块实现说明

# 基于内容推荐

## 实现过程

**一、计算Item相似度**

Item有title，area,director,starring,type,info属性，根据这些属性之间的相似度来计算Item的相似度

具体：

对area,director,type三个枚举类型属性进行匹配0/1匹配，即两个Item的这三个属性相同则相似度记为1，否则为0；

对于starring，由于starring包含多个元素，故使用按空格划分得到Item的所有star，然后计算两个Item的Starring之间的雅可比系数作为相似度；

对于Info，这是最主要的，也是最关键的。由于Info是一段对Item进行描述的文本，故使用Paoding分词得到Item的Info中出现的关键字，同时结合tf-idf与另一个Item的Info中的关键字进行计算相似度，使用的是皮尔逊算法，得到Info的相似度；

最后每个属性设置一个权重，info-0.5/star-0.1/type-0.2/director-0.1/area-0.1最终计算出每个Item与其他Item的相似度，同时将相似度值大于阈值(0.3)的存入contentbased\_item\_similarity表中

**二、计算User推荐结果**

计算推荐结果根据User评价过的Item，找出Item的相似Item作为推荐结果，存入数据库

具体：对某个User取出其所有的Rating，对于每个Rating，找出与Rating的Item的相似Item（所有），然后用Rating的value与Item的similarity求乘积作为该User对该Item可能的评分，得到User对Item的所有可能评分之后，按评分降序排列取出前topN个Item作为推荐结果存入recommender\_result表中。

**三、显示推荐结果**

根据User从推荐结果表中取出topN个对应的推荐类型结果集。

## 目前存在的问题

1. **基于内容的Item相似度推荐：**
2. 背景：主要是根据Item自身的一些信息，如地区，导演，出演明星，以及描述信息，其中描述信息是主导，辅之以其他的一些信息来进行相似度计算。
3. 问题：给的数据没有Item的信息，而且在对于测试数据的Item的描述信息进行相似度计算时发现相似度极低。
4. 原因：对于电影的描述一方面为了与众不同，另一方面描述文字出现的主要是电影中的人物角色姓名。
5. 解决方案：无

# 基于协同过滤推荐

## 实现过程--基于Item相似度的协同过滤推荐

1. 计算Item相似度：

根据Rating计算每两个Item之间的相似度，采用的是修改后的余弦相似性算法。

具体：找出共同看过这两个Item的User集合，根据User集合采用修改后的余弦相似性算法计算这两个Item的相似度。

1. 计算User的推荐结果：

根据User的评分记录，找出该User所看过的全部Item，以及对应的评分。对已每一个看过的Item，找出该Item的前10个相似Item作为可能推荐结果集。对可能推荐结果集中的每个Item预测该User对该Item的的可能评分，使用相似度\*评分，并记录对应的解释，最后根据所有预测评分排序取出20个存到到推荐结果表(recommender\_result表)。

1. 显示推荐结果：

根据User从推荐结果表中取出topN个对应的推荐类型结果集。

## 实现过程--基于**User**相似度的协同过滤推荐

1. 计算User相似度：(交集与并集的差距)

根据Rating计算每两个User之间的相似度，采用的是修改后的余弦相似性算法。

具体：找出两个User共同评过分的Item集合，根据共同的Item集合采用修改后的余弦相似性算法计算这两个User的相似度。

1. 计算User的推荐结果：

根据User的评分记录，找出该User的前十个相似用户，找出每个相似用户的评分记录，即相似用户看过的Item作为该User的可能推荐结果集，计算该User对可能结果集中的每个Item的可能评分，使用相似用户对某个Item的评分\*该User与相似用户的相似度，根据可能评分进行排序，选出20个Item作为推荐结果存入推荐结果表(recommender\_result表)。

1. 显示推荐结果：

根据User从推荐结果表中取出topN个对应的推荐类型结果集。

## 目前存在的问题

1. 背景：基于协同过滤的相似度推荐分为两部分：以及Item的协同过滤推荐以及基于User的协同过滤推荐。主要都是根据User的Rating来计算每两个User以及Item之间的相似度，然后根据每个User的Rating来寻找其推荐结果
2. 问题：协同过滤算法不准确，使用的是设计文档上的修正的余弦相似性（Adjusted Cosine），计算出的相似度结果也比较低。
3. 原因：修正的余弦相似性（Adjusted Cosine）根据两个User所评价过的所有Item进行计算余弦相似度，所以两个User之间所看过的不同的Item一会被放到一起进行计算，导致相似度降低。
4. 解决方案：取出两个User之间不同的Item，仅根据相同的Item进行相似度计算

# 基于关联规则推荐

## 实现过程

该算法主要是在Apriori算法的基础上进行了简化，按照《算法设计文档》说明，该算法只进行了二项集的计算，并没有进行K项集的计算。下面对实现的java算法进行说明。

本算法使用到数据库的交易事实表，关联推荐表，条目表，参数表。首先算法进行初始化数据，即通过交易事实表获得一项集数据，并从参数表中获取相关参数（最小支持度阈值，最小置信度阈值）；然后计算一项集的支持度，并从中剔除支持度小于最小支持度阈值的结果，剩下的一项集合即为一项频繁集；接着构造二项集，二项集的构造是将一项频繁集中的结果两两组合，并计算其支持度，剔除置信度小于最小置信度阈值的值，将剔除后的结果保存至关联推荐表。

在给用户进行推荐的时候，根据用户正在看的电影进行推荐。首先获取用户的id和正在看的电影id；根据电影id从关联推荐表中取出计算好的推荐结果，并剔除掉用户已经观看过的电影id(根据用户id从交易事实表中获取相关电影id)；如果推荐结果数量大于或等于topN(topN值从参数表中取出)，先将推荐结果进行按置信降序排序，取出前topN个结果即可；如果推荐结果数量小于topN,先按置信度值降序排序，得到第一个结果的关联结果id，在根据此id得到与其关联的结果集，并加入到推荐结果中，再次进行判断结果集的个数，如果仍然小于topN，则重复上面算法，，直至结果个数大于或等于topN。

## 目前存在的问题

上面是对于算法实现的说明，该算法目前存在的一下问题。

一是算法的时间尚待优化；

二是推荐结果集小于topN时，没有区分结果集的排序，可能导致推荐结果不是最满意结果；

# 用户兴趣模型推荐

## 实现过程

用户兴趣模型是首先依照用户的历史评分记录rating表、用户users、电影条目items表构造一张interest表，然后结合interest表与items表查询得到推荐结果。

首先是interest表的构建，通过调用RatingDaoImpl的getRatings(UserId userId)方法查找出每个用户的所有评分记录List<Rating>，然后将将Rating逐个交给service.interest包下面的UserInterestReplay类的handleItemFeedback（Rating r）方法处理。处理每条Rating的时候，可以通过Rating过去用户的id，电影条目的itemId，电影条目的评论值fbValue、确信值certainty，通过itemId获取电影条目的信息item，将fbValue、certainty、item通过用户反馈ItemFeedback类的构造方法得到ItemFeedback类的一个实例feedback。通过调用UserInterestFeedbackProcessor类的enterFeedback（UserId userId , IFeedback feedback）方法计算该type的权重，利用权重weight、用户userId、确信度certainty、兴趣类型type得到Interest类的实例，并将它存入到数据库的interest表中。

其次是结合interest表和items表查询。用户登录后，其所有的信息会保存在一个session里面，通过session得到用户的实例user，再查找该用户的兴趣模型。通过调用InterestService里面的findInterest（int parameter ，User u）得到该用户的List<InterestHibernate>，通过设置的参数，取出权重相对比较大的前几种类型的type进行推荐。通过每一种type，调用InterestService的replay（int parameter ， User user）查找出每种类型电影中平均评分最高的10部电影近推荐。

## 目前存在的问题

最后总结推荐过程中的不足之处：

1、真实的数据是没有rating，rating必须通过消费记录、浏览记录、下载记录等各种记录得到一个综合的rating；2、由于rating表里面本身就存在一个平均分，而数据的模式都是根据平均分\*一定比例+随机数\*一定比例得来的，没有参照rating表再去计算每部电影的平均分；3、计算权重时，利用的公式是 ——当前权重 + 跟新参数\*（评分值—预测值）\*类型因素 类型因素=1.0/所有的类型 预测值被设为当前权重，符合看的该类型的电影越多权重越大的情况，但是权重值设置没有上限和下限。4、还有一些类是采用的Duine里面的类，如InterestHibernate和Rating

未按比例取；剔除已经评分表中已经看过；

刘老师意见

兴趣模型需增加类型

增量更新还是重新计算