

基于项目和用户双重聚类的协同过滤推荐系统

摘要：随着互联网技术的迅猛发展，人们逐渐走入了信息过载的时代。面对大量的信息，我们都显得有些无所适从。作为信息需求者，从大量信息中找到自己感兴趣的信息往往是一件非常困难的事情；而对于信息提供者，让自己的信息脱颖而出，受到大家的关注，也是一件非常困难的事情。推荐系统就是解决这一矛盾的重要工具。推荐系统通过建立用户和信息产品之间的关系，利用已有的选择过程或相似性关系，一方面挖掘用户潜在感兴趣的信息，另一方面让信息能够展现在对它感兴趣的用户面前。协同过滤技术是目前推荐系统中应用最早和最为成功的技术之一，其发展面临着两个三个主要挑战：提高协同过滤算法的可扩展性；降低推荐系统数据集的稀疏性；提高推荐质量。针对这些问题本文提出一个改进的协同过滤算法——基于用户和项目双重聚类的协同过滤推荐算法。利用评分数据稀疏差异度和项目类别构造集合差异度度量公式，用以在用户-项目评分矩阵上进行项目聚类，解决了协同过滤推荐算法在面临高维数据稀疏特征时推荐效果差的难题；同时采用经典的 K-means 聚类方法对用户进行聚类，具有简单快速适于处理大数据的优点，省去了规范化处理。实验结果表明，此算法拥有基于用户聚类的协同过滤算法和基于项目聚类的协同过滤算法的优点，并摒弃了其缺点。

关键词：推荐系统、协同过滤、IBCRA 项目聚类、双重聚类、综合推荐

The thesis title

Abstract: With the rapid development of Internet technology, people gradually get into the age of information overload. Faced with a lot of information, we are a bit confused. As the people who need information, it's a very difficult thing to find the information they are interested in from a lot of information; while for information providers to make their message stand out is also a very difficult thing. Recommendation system is an important tool to resolve this contradiction. Recommended system through the establishment of relationships between users and information products, using the existing selection process or similar relationship, on the one hand mining information of interest to potential users, on the other hand so that information can be displayed in front of the user interested in it. Collaborative filtering technology is one of the main technologies for the recommendation system in e-commerce which has application of the earliest and is most successful. However, it has two problems: the lack of algorithm scalability and the sparsity of dataset. In order to solve these problem, we developed a hybrid recommendation system called Collaborative Filtering&User-Clustering and Item-Clustering based recommendation system. Degree of difference between the use of ratings data sparse collection of construction and project categories difference metric formula for the user - Cluster project on item rating matrix to solve the collaborative filtering algorithm in the face of high-dimensional data sparseness problem characterized by poor recommended; K-means clustering method to the user while using the classic clustering has the advantage of simple and fast for big data processing, eliminating the need for standardization of treatment. The Simulation results show that our method is more effective than traditional collaborative filtering algorithms.

Key words: Recommendation system; Collaborative Filtering; IBCRA Item-Clustering;
Double Clustering; Hybrid recommendation

目 录

1. 挖掘目标.....	3
2. 分析方法与过程.....	4
2.1. 总体流程	4
2.2. 具体步骤	7
2.3. 结果分析	14
3. 结论.....	17
4. 参考文献.....	17

1. 挖掘目标

本文通过研究已有的协同过滤技术在电子商务推荐系统应用中面临的问题和挑战，以及现有的解决算法(主要为基于用户聚类的协同过滤和基于项目聚类的协同过滤)，提出一种基于用户和项目双重聚类的协同过滤推荐算法。在进行项目聚类时，针对推荐系统数据集高维稀疏的特征，将评分数据的稀疏差异度引入到项目聚类算法中，然后将稀疏差异度和项目类别构造集合差异度相结合，对用户—项目评分矩阵进行项目聚类；而用户聚类的核心概念是假设人与人之间的行为具有某种程度的相似性，即行为类似的用户，会做出相似的选择。它通过分析用户对哪些项目进行了什么样的评价，得出用户间的相似性，从而进行预测和推荐。最后根据用户聚类 and 项目聚类的推荐结果进行加权计算，产生最终的推荐项目，缩小了比较相似性空间，提高算法的可扩展性，同时考虑用户和项目之间的类别相似性，更大程度上提高系统推荐的质量。

2. 分析方法与过程

2.1. 总体流程

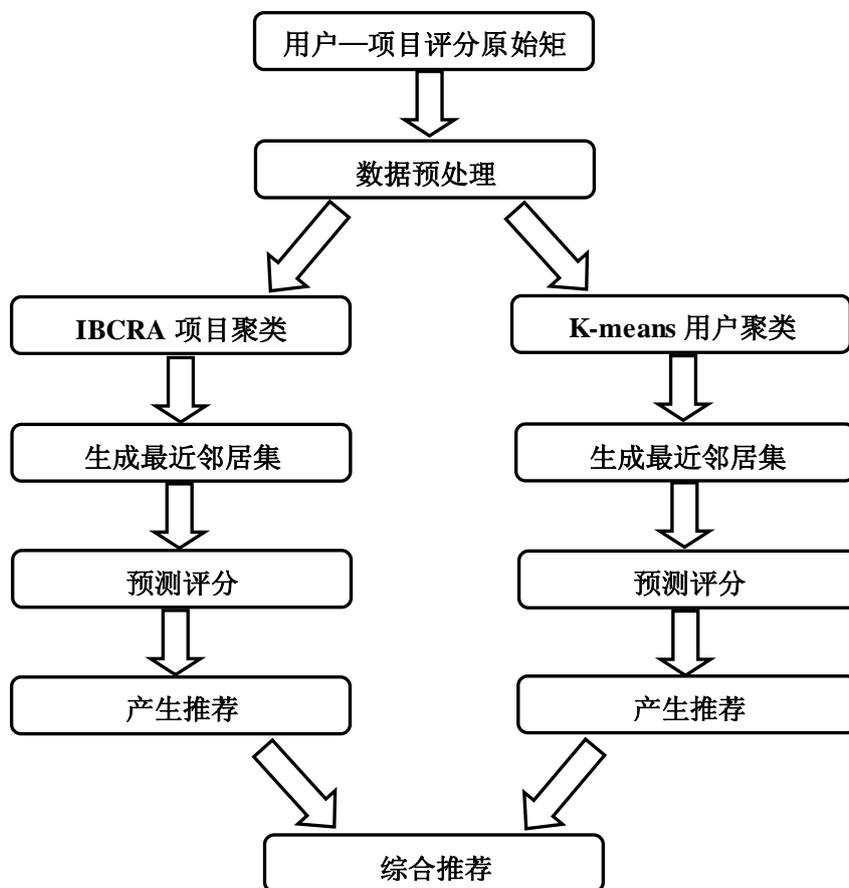


图 1 基于用户和项目双重聚类协同过滤算法流程图

主要步骤描述:

步骤一：数据预处理，将用户-项目评分矩阵进行数据预处理，作为算法输入。

步骤二：聚类，对用户-项目矩阵里所有用户进行聚类，聚成 k_1 类；对用户-项目矩阵里所有项目进行聚类，聚成 k_2 类。

步骤三：生成最近邻居集，根据目标用户与其所在的类别中用户的相似度生成最近邻居集；根据目标项目与其所在的类别中项目的相似度生成最近邻居集。

步骤四：预测评分，在利用用户相似度生成的最近邻居集中，计算未评价项目的预测值；在利用项目相似度生成的最近邻居集中，计算未评价项目的预测值。

步骤五：综合推荐，选取基于用户的协同过滤算法预测出来的，评分值最高的前 n_1 个项目；选取基于项目的协同过滤算法预测出来的，评分值最高的前 n_2 个项目 ($n_1=n_2$)。结合用户聚类 and 项目聚类产生的推荐集 n_1 和 n_2 ，按照预测值从高到低，选取前 n 项，若项相同则加权求和作为预测评分。

2.2. 具体步骤

步骤一：数据预处理

(1) 数据来源

本文选用的实验数据集为 MovieLens 数据集 (<http://MovieLens.umn.edu/>)，是被使用最为广泛的实验数据，并且已经成为评价推荐算法的基准数据集。MovieLens 站点用于接收用户对电影的评分并提供相应的电影推荐列表，其评分尺度是从 1 到 5 的整数，数值越高，表明用户对该电影的偏爱程度越高。

(2) 数据简化

本文的实验数据集是在 MovieLens 数据集的全部 100000 条数据中，筛选去用户编号小于 500 (用户编号的取值范围在 1 到 943)、项目编号小于 1000 (项目编号的取值范围在 1 到 1682)，共得到 52153 条评分数据。

(3) 数据集的划分

交叉验证法^[1]是一种没有任何前提假定直接估计泛化误差的模型选择方法，由于没有任何假定，可以应用于各种模型选择中，因此具有应用的普遍性，又由于其操作的简便性。

实验采用 5 折交叉验证法，将实验数据集平均分成 5 个互不相交的数据子集，训练集和测试集的数据比例为 4:1。每次实验选择其中一个数据子集作为测试集，其余四个数据子集作为训练集。如此循环五次，取每次实验结果的平均值作为最终结果。5 折交叉验证法可以有效降低数据集对实验结果的影响。在 MovieLens 数据集中，评分数据是按照用户编号从低到高升序排列，同一用户的评分数据按照项目编号升序排列。在对实验数据进行划分时，为了保证不因实验者的偏好或数据集

本身差异造成实验结果的失真，采用对评分数据打标记的方式。标记位的取值为 1 到 5 的整数，对于实验数据集的第一条评分数据标记为 1，第二条评分数据标记为 2，依次递增。当上一条评分数据的标记位已经取值为 5 时，下一条评分数据的标记位重新取值为 1，直到为实验数据集全部数据打标截止。根据打标后生成的标志位结果，对数据集按照不同的组合进行分类，得到下表：

表 1 数据集划分方法

数据集	标记位	数据条目
train1	1 2 3 4	41720
test1	5	10433
train2	1 2 3 5	41722
test2	4	10431
train3	1 2 4 5	41723
test3	3	10430
train4	1 3 4 5	41721
test4	2	10432
train5	2 3 4 5	41722
test5	1	10431

A. IBCRA 项目聚类

步骤二：聚类

(1) IBCRA 项目聚类算法^[2]：

一方面，用户—项目评分矩阵 R 是典型的高属性维稀疏矩阵。高属性维稀疏数据对象间的稀疏相似性可以通过计算对象间的稀疏特征的差异度来描述，差异度越大，对象越不相似；差异度越小，对象越相似，即差异度反映对象间的相似程度。另一方面，项目类别在一定程度上可以反映用户消费的偏好，如果用户对某类别的商品给以较高的评分，则其很有可能给同一类别的其他项目相应的评分。如果两个项目属于的项目类别越多，它们之间的相似性就应该越高。从项目聚类的角度，项目类别是一个天然的聚类依据，通过项目类别可以迅速将项目集划分为若干类。

(2) 算法思想

IBCRA 算法聚类次序按照各项目评分用户数目由多到少依次聚类，其它的一些基于聚类的协同过滤推荐算法研究文献也证明在项目聚类过程中，评分用户多的项目首先开始聚类有益于提高聚类质量。如王辉等^[3]在使用 K-means 算法对评分数据集进行用户聚类时，以访问量（有效数据）最多的 k 个用户作为初始的 k 个聚类中心，经过实验验证可以较好地减少孤立点。Quan 等设计的 GroupByUSimStability 项目聚类算法，就是将项目集按照评分用户个数由多到少排列，选择评分用

户数排在前 10%的项目首先开始聚类。

本文的项目聚类算法在已调整好次序的数据集上对评分数据稀疏差异度和项目类别构造集合差异度的度量指标进行计算，通过集合差异度与阈值的比较判定两个项目集合是否可以归为一类。算法采用“自底而上”的层次聚类法，将每一个项目作为一个类开始，依照项目次序逐个向上聚集，合并最相似的项目。

(3) IBCRA 项目聚类算法主要步骤:

输入：用户—项目评分矩阵 R ，项目类别矩阵 S_{item} ，项目类别因子 I ，项目评分因子 K ，集合差异度阈值 d ，项目类别总数 G 。

输出：项目聚类簇 $Cluster$ 。

1. 设项目集 N 中项目总数为 n ，依次计算项目 I_1, I_2 直至 I_n 评分用户数目，并按照评分用户数目从大到小排列，项目重新排列后得到的项目集记作 N' 。
2. 首先为项目集 N' 中的每一个项目创建一个初始项目集合 $C_i^{(0)}$ ， $i \in \{1, 2, \dots, n\}$ 。每个项目集合中只有一个项目，如项目集 $C_1^{(0)}$ 对应项目集 N' 中的第一个项目。
3. 计算项目集合 $C_1^{(0)}$ 与项目集合 $C_2^{(0)}$ 的集合差异度，计算公式为：

$$SFD(C_1^{(0)}, C_2^{(0)}) = \frac{K - I}{K \times G}$$

如果 $SFD(C_1^{(0)}, C_2^{(0)})$ 小于集合差异度阈值 d ，则将项目集合 $C_1^{(0)}$ 与 $C_2^{(0)}$ 合并，合并后的新项目集合记作 $C_1^{(1)}$ ；如果 $SFD(C_1^{(0)}, C_2^{(0)})$ 超过集合差异度阈值 d ，则将 $C_1^{(0)}$ 与 $C_2^{(0)}$ 作为两个新的项目类，新的项目类记作 $C_1^{(1)}$ 与 $C_2^{(1)}$ 。项目类的个数记作 m 。

4. 对于项目集合 $C_3^{(0)}$ ，计算 $SFD(C_3^{(0)}, C_2^{(1)})$ ， $i \in \{1, \dots, m\}$ 寻找 i_0 ，使得

$$SFD(C_3^{(0)} \cup C_{i_0}^{(1)}) = \min_{i \in \{1, \dots, m\}} SFD(C_3^{(0)} \cup C_i^{(1)})$$

如果 $SFD(C_3^{(0)} \cup C_{i_0}^{(1)})$ 小于集合差异度阈值 d ，则将 $C_3^{(0)}$ 与 $C_{i_0}^{(1)}$ 合并，新类仍然计作 $C_{i_0}^{(1)}$ ；反之，则将 $C_3^{(0)}$ 作为一个新类，记作 $C_{m+1}^{(1)}$ ； $m+1=>m$ 。

5. 对于集合 $C_j^{(0)}$ ， $j \in \{4, 5, \dots, n\}$ ，依次重复步骤（4），进行相应的类别划分。
6. 通过项目集合的合并后得到的项目集合构成项目聚类簇 $Cluster$ 。项目聚类簇中包括 m 个项目类，每个项目类中记录了属于该项目类的项目编号。将孤立点的类从终类中去除，得到的即为最终聚类结果。

步骤三：生成最近邻居集

目标项目的最近邻居项目集合 M_i 的形成主要取决于项目相似性计算方法。不同的项目相似性计算方法会影响项目与目标项目的相似性计算结果，进而影响最近邻居项目集合内项目排列顺序和权重。加权平均法根据用户对目标项目邻居项目集合内的项目评分加权求和得到，权重由邻居项目同目标项目的相似度构成。用户 u 对项目 i 的预测评分 $r_{u,i}$ 的计算公式为

$$r_{u,i} = \frac{\sum_{j \in M_i} sim(i, j) \times R_{u,j}}{\sum_{j \in M_i} |sim(i, j)|}$$

其中， $R_{u,j}$ 表示用户 u 对项目 j 的评分， $sim(i, j)$ 表示项目 i 和项目 j 的相似性， M_i 表示目标项目 i 的最近邻居项目集合。

生成最近邻居集的流程如图 2 所示，最近邻居集搜索的主要目标就是为目标用户 u 寻找出与其相似度最高的若干个双聚类，组成目标用户 u 的最近邻居集。

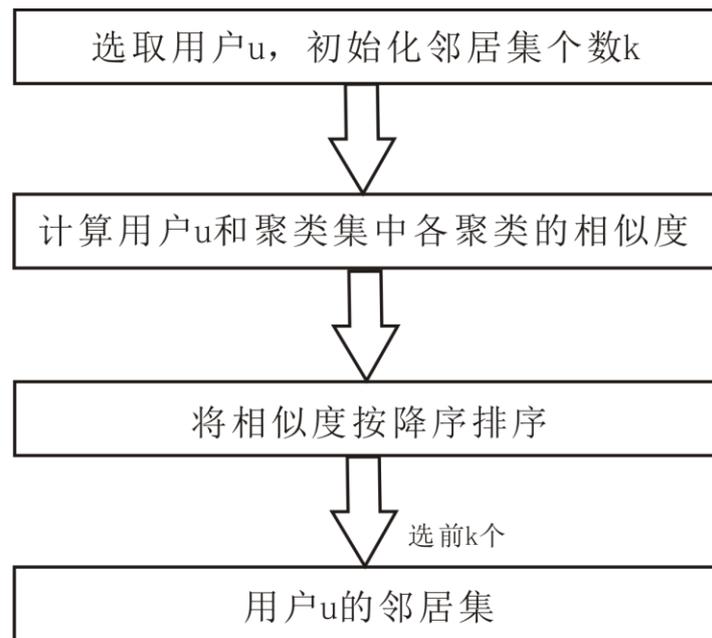


图 2 生成项目聚类的最近邻居集

步骤四：预测评分

1. 针对目标项目 i ，在项目聚类簇 Cluster 寻找目标项目 i 所属于的项目类 index。
2. 设 $I = \{i_1, i_2, \dots, i_h\}$ 为推荐系统项目类 index 中全部项目的集合，则在集合 I 内计算项目 i 和项目 j ($j \in I$ 且 $j \neq i$) 的相似性，项目相似性计算使用余弦相似性公式。

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\sum_{u \in U} R_{u,i} \cdot R_{u,j}}{\sqrt{\sum_{u \in U} R_{u,i}^2 \cdot \sum_{u \in U} R_{u,j}^2}}$$

3. 如果目标项目 i 为孤立点, 或者目标用户 u 对项目类 index 类内其他项目都没有评分, 则在全体项目集内计算项目 i 与其他项目的相似性。
4. 统计目标用户 u 在 index 类中已经评分的项目的集合 I_u ; 如果 i 为孤立点, 则集合 I_u 为目标用户 u 所有已经评分的项目的集合。
5. 计算目标用户 u 的未评分项目集, 即没有被该用户评价的项目组成的集合 $I_u' = N - I_u$, 其中 N 表示全体项目集合。
6. 根据计算得到的项目相似性计算结果和输入的项目最近邻居个数 K_{near} , 计算目标项目 i ($i \in I_u'$) 的 K_{near} 个最近邻居, 邻居项目组成集合 $M = \{i_1, i_2, \dots, i_{K_{\text{near}}}\}$, 相似度 $\{sim(i, i_1), sim(i, i_2), \dots, sim(i, i_{K_{\text{near}}})\}$ 从大到小的顺序排列。如果目标用户 u 已评分的项目总数小于参数 K_{near} , 则集合 M_i 只选择 $\text{Num}(I_u)$ 个最近邻居, $\text{Num}(I_u)$ 表示项目类 index 中项目总数。
7. 根据步骤 (6) 得到的目标用户 u 的最近邻居集合 M_i 和用户—项目评分矩阵 R 内的评分数据, 依据用户 u 对目标项目 i 的最近邻居评分的加权平均值预测用户 u 对项目 i ($i \in I_u'$) 的预测评分, 即

$$r_{u,j} = \frac{\sum_{j \in M_i} sim(i, j) \times R_{u,j}}{\sum_{j \in M_i} |sim(i, j)|}$$

8. 重复步骤 (6) 和步骤 (7), 预测目标用户 u 对所有的未评分项目的评分, 选择评分最高的前 n_1 个项目推荐给用户。

B. K-means 用户聚类

步骤二: 聚类

(1) K-means 算法的基本思想:

K-means 算法的基本思想是首先从含有 n 个数据对象的数据集中随机选择 K 个数据对象作为初始中心，然后计算每个数据对象到各中心的距离，根据最近邻原则，所有数据对象将会被划分到离它最近的那个中心所代表的簇中，接着分别计算新生成的各簇中数据对象的均值作为各簇新的中心，比较新的中心和上一次得到的中心，如果新的中心没有发生变化，则算法收敛，输出结果，如果新的中心和上一次的中心相比发生变化，则要根据新的中心对所有数据对象重新进行划分。直到满足算法的收敛条件为止^[4]。

K-means 算法具有过程简单易理解、快速有效、适于处理大数据集等优点，从以上角度出发，本文选取了 K-means 对用户集聚类^[7]。

(2) K-means 算法的主要工作流程图如图 3 所示:

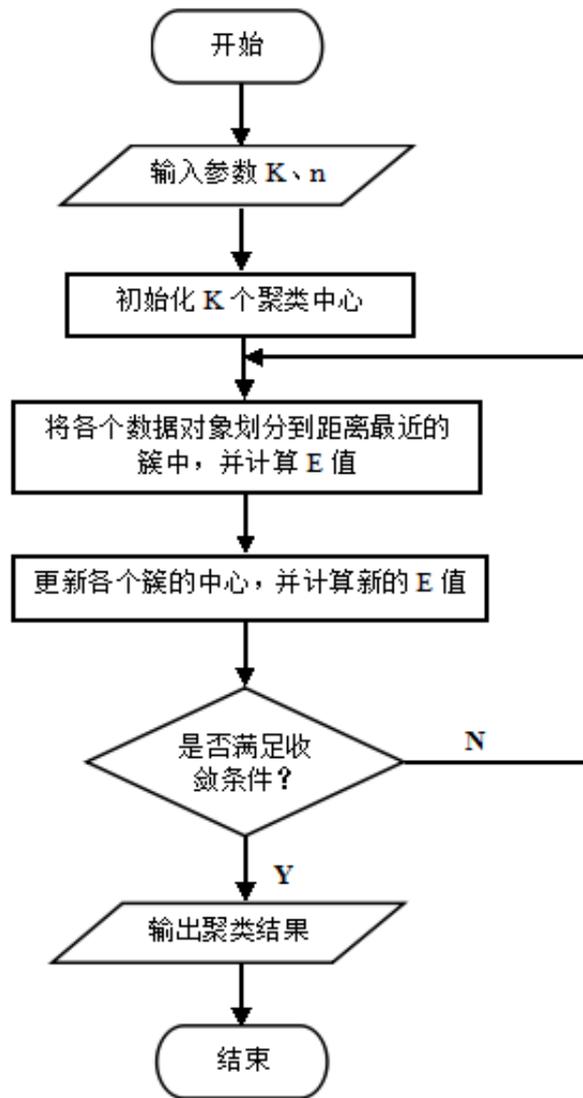


图 3 K-means 算法的主要工作流程图

(3) K-means 算法的伪代码

1. 在所有用户中随机挑选 K 个用户作为初始聚类中心点;
2. repeat;
3. 计算剩余的其他用户与各个聚类中心的相似度, 并根据相似度将相应用户分配到对应的类簇;
4. 计算上一步新生成的类簇的中心点;
5. when 各个类簇的中心点不再产生变化
end。

步骤三：生成最近邻居集

邻居集是要向目标用户产生推荐的用户集合, 是一个与目标用户兴趣最相近的用户集合。

邻居集产生步骤: (生成最近邻居集的流程图如图 4 所示)

- (1) 在聚类簇中找到目标用户所在的类簇;
- (2) 计算该类簇中其它用户与目标用户综合相似度最高的 K 个用户;
- (3) 将这 K 个用户归类为目标用户的最近邻居集。

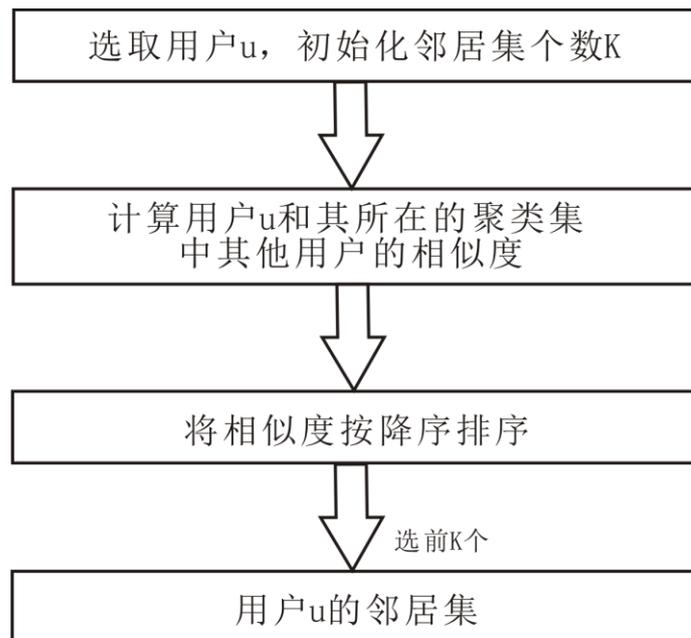


图 4 生成用户聚类的最近邻居集

步骤四：预测评分

在最近邻居集中进行目标用户评分的预测, 计算出目标用户所有未评分项目的预测评分, 将这些项目按照预测评分从大到小排序, 选取其中最大的 n_2 个项目推荐给目标用户。

相关公式如下:

$$P_{a,y} = \bar{r}_a + \frac{\sum_{i \in N_a} sim_{a,i} \times (r_{iy} - \bar{r}_i)}{\sum_{i \in N_a} |sim_{a,i}|}$$

$P_{a,y}$ 表示目标用户对未评分项目 i_y 的预测评分， N_a 表示目标用户的最近邻， \bar{r}_a 表示目标用户的平均评分， \bar{r}_i 表示用户 u_i 的平均评分

步骤五：综合推荐

结合项目聚类 and 用户聚类通过双重聚类的方法^[5]产生推荐，对于目标用户 i ，通过用户聚类产生 Top-n 推荐 n_1 ，通过项目聚类产生 Top-n 推荐 n_2 ，如果两推荐中有相同项目，则加权求和，最后选择预测值最大的 n 个项目组成最终推荐项目集合。

推荐公式： $sim = percent * sim_1 + (1 - percent) * sim_2$

其中，Sim：最终的评分；percent：所占权重； sim_1 、 sim_2 分别为基于项目、用户的预测评分。

综上所述，本文的整体算法框图如图 5 所示

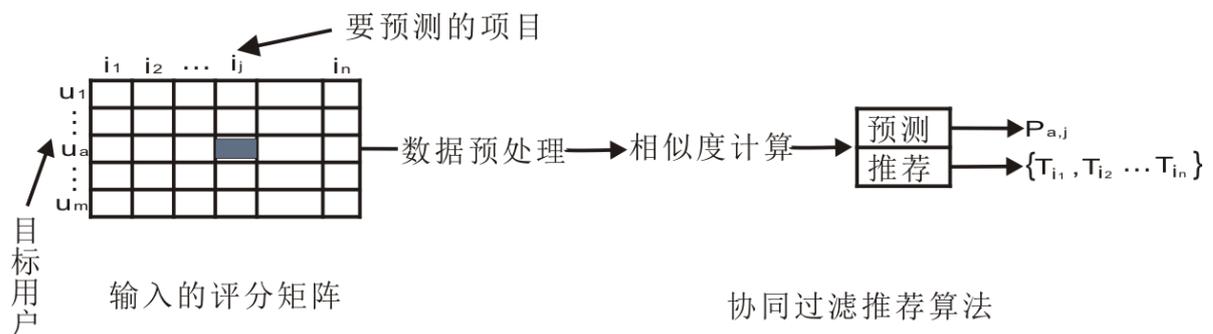


图 5 整体算法框图

2.3. 结果分析

本文共使用了三种评价标准来评价推荐质量，除了最常用的平均绝对误差 MAE 外，还选取了平均平方误差 RMSE 以及 F1 作为评价标准。MAE^[8] 是预测值和实际值的平均绝对误差，它可用来衡量预测值的准确性，可以比较直观地对推荐质量进行度量，MAE 值越小表示推荐质量越高；RMSE^[6] 是对 MAE 的改进，它可以减小极端值对算法效果的影响，RMSE 值越小表示推荐质量越高；F1 是通常的准确率和召回率两种评估标准的综合^[10]，用来对项目推荐的准确性和全面性进行评估，F1 值越大表明推荐的质量越高。

(1) MAE 指标的对比结果：

MAE 是预测值和实际值的平均绝对误差，它可用来衡量预测值的准确性，可以比较直观地对推荐质量进行度量，MAE 值越小表示推荐质量越高，定义式如下：

$$MAE = \frac{1}{N} \sum_{i=1}^N |p_i - q_i|$$

其中: N 为项目数， p_i 为预测的评分， q_i 为实际评分。

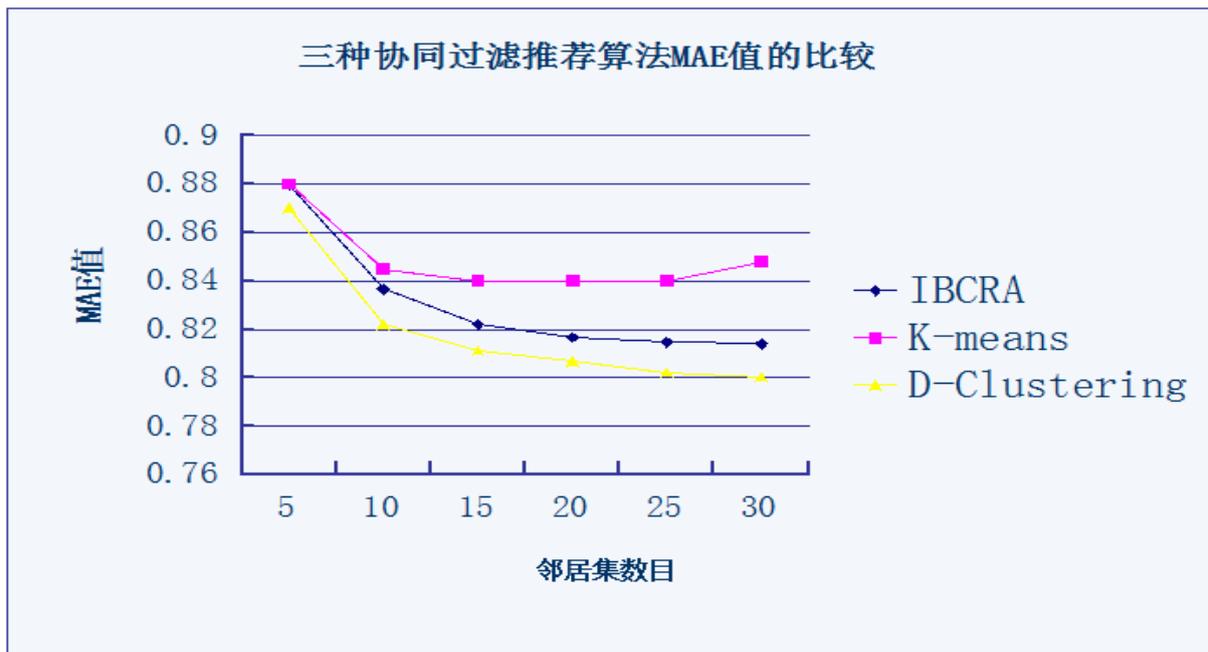


图 6 三种协同过滤推荐算法 MAE 值的比较

(2) F1 性能比较:

在推荐系统评价指标中，准确率（Precision）、召回率（Recall）是一种经常用到的指标^[9]：

准确率（Precision）表示目标用户对推荐给他的项目感兴趣的概率：

$$R_p = \frac{n_{TP}}{n_{TP} + n_{FP}}$$

n_{TP} 表示算法推荐且用户喜欢，正确推荐的项目数； n_{FP} 表示算法推荐但用户不喜欢，错误推荐的项目。

召回率（Recall）表示用户喜欢的商品被推荐的概率：

$$R_r = \frac{n_{TP}}{n_{TP} + n_{TN}}$$

n_{TP} 表示算法推荐且用户喜欢，正确推荐的项目数； n_{TN} 表示算法不推荐且用户不喜欢，正确不推荐的项目数。

准确率与召回率往往是负相关的，因此综合两者，研究者们提出了 F_1 指标，定义如下：

$$F_1 = \frac{2 \cdot R_p \cdot R_r}{R_p + R_r}$$

F_1 越大，推荐质量越高。

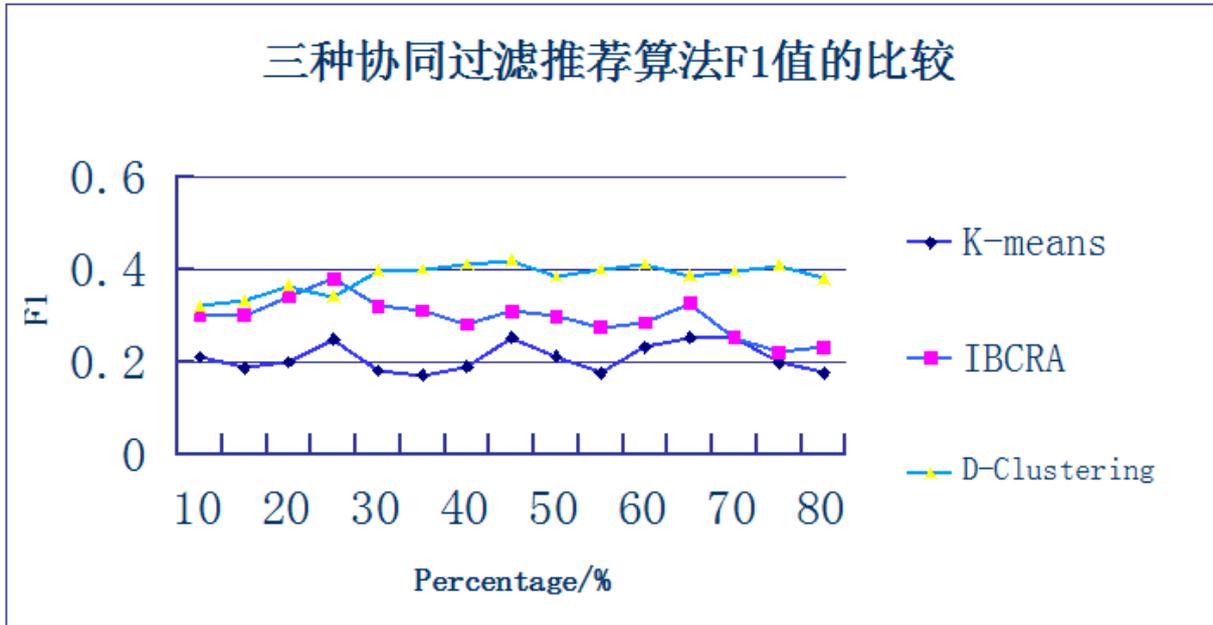


图 7 三种协同过滤推荐算法 F1 值的比较

图中的实验结果表明，D-Clustering 模型在不同的 Percentage 取值下，相比其他模型在 F1 值和表现稳定性方面具有明显的优势。当 D-Clustering 模型在 MAE 方面具有优势时，算法能够更加精准的为用户预测未评分项目的评分，这样便可以为用户推荐真正喜欢的项目，从而在一定程度上提高了 Precision 和 Recall 方面的性能，而这表现为 F1 方面的提升。

(3) RMSE 值的对比结果：

平均平方误差 RMSE(Root Mean Squared Error) 也是一种推荐质量的评价标准，其定义如下：

$$RMSE = \sqrt{\frac{\sum_{i=1}^N |p_i - q_i|^2}{N}}$$

其中:N 为项目数， p_i 为预测的评分， q_i 为实际评分。RMSE 越小，说明推荐质量越好。用户邻居数取 35，目标用户的最近邻居数取 10， 15， 20， 25， 30， 35。

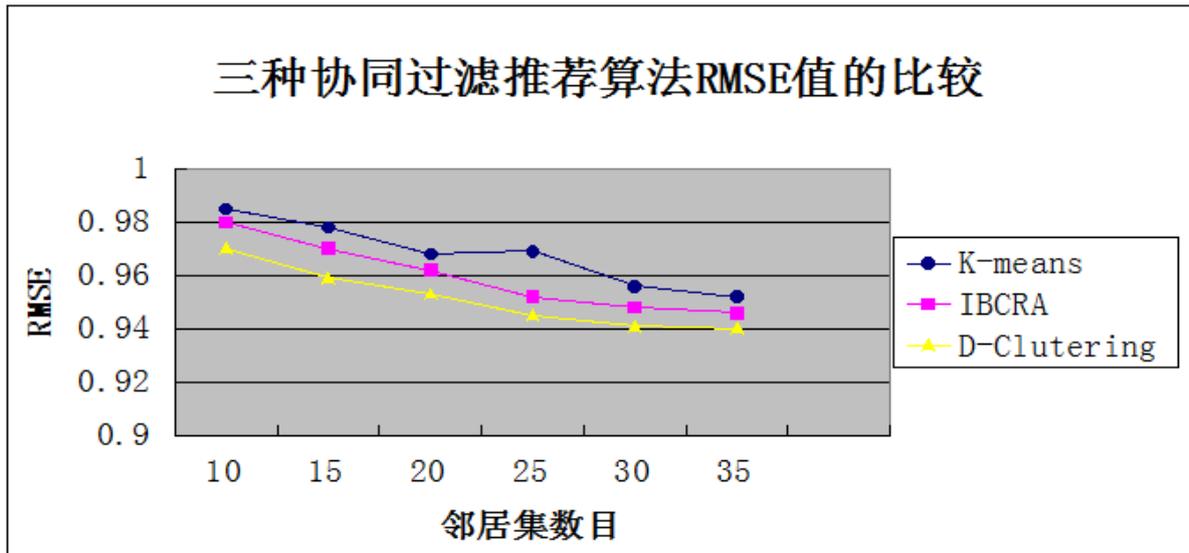


图 8 三种协同过滤推荐算法 RMSE 值的比较

从图中可以清楚地看出，无论是 K-means 用户聚类算法还是 IBCRA 项目聚类，RMSE 值都高于本文提出的 D-Clustering 聚类算法，而且随着邻居集数目的增加，推荐效果越来越好。

综合以上结果分析可知，无论以哪种评价标准来衡量推荐质量，本文提出的基于用户和项目双重聚类的协同过滤推荐算法都要高于单方面的基于用户聚类和基于项目聚类的协同过滤推荐算法。

3. 结论

本文提出的基于用户和项目双重聚类的协同过滤推荐算法，首先进行项目聚类，项目聚类中使用了基于项目的协同过滤推荐算法预测未评分项目评分，因而数据稀疏度问题也得到了一定程度上的解决，同时也解决了基于项目聚类的协同过滤算法无法进行“跨类型推荐”的难题，同时进行用户聚类，并综合用户类和项目类，寻找最近邻居集，其考虑了用户之间和项目之间的类别相似性，因而其推荐质量要高于单方面的基于用户聚类和基于项目聚类的协同过滤推荐算法。最近邻居是在用户聚类和项目聚类类内生成的，因而其推荐速度快，一定程度上解决了可扩展性(实时性)问题，同时结合两种推荐算法的推荐结果，使推荐质量更高。

4. 参考文献

- [1] 范永东. 模型选择中的交叉验证方法综述[D]. 山西: 山西大学, 2013.
- [2] 姚忠, 魏佳, 吴跃. 基于高维稀疏数据聚类的协同过滤推荐算法[J]. 信息系统学报, 2008.
- [3] 王辉, 高利军, 王听忠. 个性化服务中居于用户聚类的过滤推荐[J]. 计算机应用, 2007.
- [4] 欧陈委. K-均值聚类算法的研究与改进[D]. 湖南: 长沙理工大学, 2011.
- [5] 施华. 基于项目和用户双重聚类的协同过滤推荐算法[D]. 吉林: 东北师范大学, 2009
- [6] 李克潮, 凌霄娥. 云模型与用户聚类的个性化推荐[J]. 计算机应用, 2013.
- [7] 王宗武. 基于用户聚类的协同推荐算法研究[D]. 广东: 广东工业大学, 2013.
- [8] 季昀. 基于协同过滤推荐算法电影网站的构建[D]. 哈尔滨工业大学, 2009.
- [9] 康美林. 基于双聚类模型的协同过滤推荐引擎研究[D]. 中南林业科技大学, 2013.
- [10] 杨新耀, 于炯, 吐尔根·伊布拉音, 廖彬. 综合用户和项目预测的协同过滤模型[J]. 计算机应用, 2013.