# 基于用户偏好和项目属性的协同过滤推荐算法①

姚平平, 邹东升, 牛宝君

(重庆大学 计算机学院, 重庆 400044)

**摘 要**: 协同过滤推荐算法是目前应用最为广泛的个性化推荐方法之一, 但传统的推荐算法在计算目标用户邻居集时只考虑用户项目评分矩阵中的具体数值, 没有考虑用户偏好以及用户评分与项目属性之间的关系, 推荐精度也有待进一步提高. 针对这一问题, 提出了一种基于用户偏好和项目属性的协同过滤推荐算法(UPPPCF). 本算法在传统的用户项目评分矩阵基础上综合考虑用户偏好以及项目属性, 把评分矩阵转变成基于用户偏好的用户项目属性评分矩阵, 然后根据这一评分矩阵来计算目标用户的最近邻居集, 克服了传统相似性计算方法只依靠用户评分值的不足, 同时本文对预测值判定给出了一种有效的度量方法. 在 MovieLen 数据集上的实验结果表明, 本文提出的 UPPPCF 算法能够有效弥补传统协同过滤算法中的不足, 而且在推荐精度上有了明显的提高.

**关键词**: 协同过滤; 推荐系统; 用户偏好; 用户项目属性评分矩阵

## Collaborative Filtering Recommendation Algorithm Based on User Preferences and Project Properties

YAO Ping-Ping, ZOU Dong-Sheng , NIU Bao-Jun

(College of Computer Science, Chongqing University, Chongqing 400044, China)

**Abstract**: Collaborative filter algorithm is one of the most widely used technologies of personalized recommendation. However, the existing recommendation algorithms only consider the user item rating matrix specific value while calculating the target user neighbor. User preferences and user ratings and the relationship between the project properties are ignored. Moreover, the accuracy also needs to be further improved. To solve this problem, this paper proposed a new collaborative filtering algorithm based on user preferences and project properties (UPPPCF). By using the traditional user project evaluation matrix, the algorithm synthesizes user preferences and the project properties. The project score matrix is changed into project properties score matrix based on user preference. Then the nearest neighbors of target user sets are computed according to this new score matrix. As a result, the proposed algorithm overcomes the insufficiency of existing similarity calculation methods, which only depend on user ratings value. Meanwhile, an effective measurement method for predictor decision is suggested in this paper. The experimental results on MovieLen datasets show that the proposed algorithm can effectively improve the existing traditional collaborative filtering. In addition, the recommendation accuracy has been significantly improved.

**Key words**: collaborative filtering algorithm; recommendation systems; user preferences; user project properties rating matrix

互联网技术的快速发展将人类带入了一个崭新的信息时代, 尤其是电子商务对整个社会的发展和个人的生活都产生了巨大的影响, 随着电子商务规模的不断扩大, 商品种类和数量快速增长, 顾客需耗费大量时间从海量数据中寻找所需的商品, 潜在的消费者往往因淹没在海量信息中而流失, 这就是信息超载[1].

① 基金项目:国家自然科学基金(61309013);重庆市基础与前沿研究计划(cstc2014jcyjA40042)
收稿时间:2014-11-07;收到修改稿时间:2014-12-05

推荐系统正是为了解决这一问题, 它主要是用来帮助人们能够更快更准确的挑选出自己最感兴趣的商品, 推荐结果的准确性时决定推荐系统成败的关键因素, 如果系统向用户推荐的商品并不是其所需要的, 那么用户就会对推荐系统失去信心, 把推荐信息作为垃圾信息. 为了提供精确快速的推荐,研究者提出了多种推荐算法, 主要包括基于内容的推荐、基于关联规则的推荐、基于用户统计信息的推荐、基于协同过滤的推荐等, 其中协同过滤推荐算法是目前应用最为广泛的个性化推荐方法[2].

协同过滤推荐算法通过参考与目标用户具有相似兴趣和需求的其他用户的选择来决定如何为该用户进行信息推荐. 其主要思想是: 如果两个用户对项目中的一些项目的评分比较相似, 则他们对其他项目的评分也比较相似. 协同过滤推荐技术的应用领域非常广泛, 其中 Tapestry[3]是最早的推荐系统之一, 该系统记录了每个用户对他们阅读文章的观点, 并且这些观点可以被其他用户进行获取. GroupLens/Net Perceptions[4]、Ringo/Firefly[5]以及 MovieLens 都是较早期的著名的推荐系统. 协同过滤推荐算法最大的优点是对推荐对象没有特殊要求, 能够处理非结构化的复杂对象, 如书籍、文章、网页、音乐、电影以及百货等. 虽然协同过滤推荐系统有很多优点, 不过也存在着一些问题, 例如说稀疏数据、冷启动以及扩展性问题等.

随着对协同过滤推荐算法研究的不断深入, 许多研究者提出了一些新的方法来改进传统协同过滤算法的不足. 黄创光[6]等人提出了一种不确定近邻的协同过滤推荐算法, 在不确定的场景中, 结合用户以及产品的推荐结果, 通过不确定近邻因子及调和参数去计算基于用户和产品的预测评分并产生推荐. 该算法能够缓解用户评分数据的稀疏性问题并且提高了推荐质量. 陈健[7]等通过建立 k 最近邻及其影响集来预测的评分; Rajasangari[8]将基于内容的方法和基于协同过滤算法结合, 缓解了协同过滤推荐算法面临的数据稀疏和冷启动问题从而提高了推荐质量; 吴湖[9]等提出了一种两阶段评分预测方法, 大幅度降低预测阶段计算量的同时提高非负矩阵分解算法在面对数据稀疏预测上的准确度; 陶俊[10]等提出了一种基于用户兴趣分类的协同过滤推荐算法, 该算法旨在提出了适应用户兴趣多样性的协同过滤算法并利用改进的模糊聚类算法搜索最近邻来改善推荐算法的准确性.

上述算法都只关注用户项目评分矩阵中的评分值, 并没有考虑用户偏好以及用户评分与项目属性之间的关系对推荐精度的影响. 实践证明直接使用用户项目评分矩阵来计算目标用户的邻居集并不是完全符合用户选择项目的实际情况, 如电影《特种部队 2: 全面反击》的属性为科幻、惊悚、冒险和动作, 但是单独的用户项目评分值并不能体现用户真正的偏好所在, 因此仅根据用户项目评分信息进行推荐的系统通常会得不到理想的推荐结果甚至是错误的推荐结果. 为了解决这一问题, 王义[11]等提出了基于用户行为的个性化推荐系统的设计与应用, 该方法主要采用前向融合组合推荐策略, 这种前向融合技术, 能够通过缩小协同过滤的输入规模, 减少用户评分矩阵纵向的深度, 将评分矩阵中用户之间没有任何关系, 转变为具有一定相似性的用户集合, 来提高用户相似性度量的准确性. 严冬梅[12] 等提出的基于用户兴趣点和特征的优化协同过滤推荐算法, 该算法首先通过计算用户对项目的兴趣度来对用户进行分组, 然后采用贝叶斯算法计算出用户具有不同特征时对项目的喜好程度, 并且使用了一种新的相似度方法计算目标用户的最近邻居集来提高推荐精度. 虽然能够一定程度的推荐质量, 但要人工地确定阀值, 这样使用起来既不方便又不够客观. 嵇晓声[13]等提出了一种基于用户兴趣度的相似性度量方法, 该方法利用用户对不同项目类别的兴趣程度与用户评分相结合进行用户之间的相似性计算. 刘芳先[14]等提出的改进的系统过滤推荐算法进行比较, 该算法在计算用户的相似性时增加了项目相关性, 同时在预测评分过程中引入了时间函数, 将随用户兴趣的转移发现用户最近的兴趣来进行推荐, 提高推荐精度.

为了更好地解决上述问题, 本文提出了一种基于用户偏好和项目属性的协同过滤推荐算法(Collaborative filtering recommendation algorithm based on user preferences and project properties), 简称 UPPPCF. 算法综合考虑了用户偏好、项目属性, 克服了传统相似性计算方法只依靠用户评分值进行相似性计算方法的不足. 同时提出了一种新的预测值判定方法, 该方法对用户的偏好进行区分, 除去偶然评分对用户评分趋势的影响, 从而得到一个更为准确的评分, 最后进行推荐.

## 1 问题定义及基本方法

### 1.1 基于用户的协同过滤算法(UBCF)

UBCF 与人们传统的口头传达的行为习惯最为相似, 它的产生来源于对生活经验的假设: 如果两个用户的兴趣偏好相似, 那么这两个用户对同一商品的喜好也会很相似. 正因为这样, 一个用户喜好的商品就可以作为推荐给另一个用户,主要有以下三个步骤:

1) 数据表述

数据表述主要是对数据集的描述, 通常表述为一个 m×n 的用户项目评分矩阵 R, 其中 m 代表用户数, n 代表项目数, $R_{ij}$ 表示第 i 个用户对第 j 个项目的评分值, 一般 $R_{ij}\in[0,5]$其中 1 表示不喜欢, 5 表示非常喜欢, 如果没有对项目进行评分那么就为 0.

2) 计算最近邻居集

这个阶段是寻找目标用户的最近邻居集, 该步骤也是整个推荐算法中最为重要的一步, 如果查询到的最近邻居集和目标用户非常相似, 对应的推荐更准. 计算相似性的常规方法主要有三种: 余弦相似性[15]、修正的余弦相似性以及相关相似性(Pearson 相关)[4].

① 余弦相似性: 这种方法计算用户之间的相似度时, 用户被认为是一个 n 维向量, 向量的值也就是用户对项目的评分. 用户 a 和用户 b 的相似度就是两个向量夹角的余弦值. 余弦值越大, 表明两个用户的相似度越大. 余弦相似性计算的公式如下:

$$sim(a,b)=\cos(\vec{a},\vec{b})=\frac{\vec{a}\bullet\vec{b}}{\|\vec{a}\|\bullet\|\vec{b}\|} \tag{1}$$

② 修正的余弦相似性: 余弦相似性度量方法没有考虑不同用户的评分尺度问题, 修正的余弦相似性度量方法通过减去用户对项目的平均分来改善上述缺陷. 引入用户平均评分的相似性计算公式为:

$$sim(a,b)=\frac{\sum_{j\in I_{ab}}(R_{a,j}-\bar{R}_a)(R_{b,j}-\bar{R}_b)}{\sqrt{\sum_{j\in I_a}(R_{a,j}-\bar{R}_a)^2}\sqrt{\sum_{j\in I_b}(R_{b,j}-\bar{R}_b)^2}} \tag{2}$$

其中 $I_{ab}$ 代表了用户 a 和 b 都评分的项目, $I_a$ 代表用户 a 评分的项目, $I_b$ 代表用户 b 评分的项目, $\bar{R}_a$ 表示的是用户 a 的平均评分, $\bar{R}_b$ 表示的用户 b 的平均评分.

③ 皮尔森(Pearson)相关相似性: 这种相似度计算方法是基于相关性的. 这种方法在计算时, 首先找到两个用户共同评分的项目集合, 然后基于此集合计算两个用户向量的相关系数. 皮尔森相关系数相似性计算公式这样定义:

$$sim(a,b)=\frac{\sum_{j\in I_{ab}}(R_{a,j}-\bar{R}_a)(R_{b,j}-\bar{R}_b)}{\sqrt{\sum_{j\in I_{ab}}(R_{a,j}-\bar{R}_a)^2}\sqrt{\sum_{j\in I_{ab}}(R_{b,j}-\bar{R}_b)^2}} \tag{3}$$

3) 预测推荐项目

通过相似邻居集中用户的兴趣对目标用户进行推荐, 计算公式如下:

$$P(a,j)=\bar{R}_a+\frac{\sum_{i\in UN_a}sim(a,i)\times(R_{i,j}-\bar{R}_i)}{\sum_{i\in UN_a}sim(a,i)} \tag{4}$$

其中, $P(a,j)$ 代表了目标用户 a 对项目 j 的预测值, $UN_a$ 则表示目标用户 a 的最近邻居集合. sim(*a*,*i*) 是目标用户 a 与邻居集中的用户 i 的相似值.

### 1.2 基于用户协同过滤算法的不足

由 1.1 节可知, UBCF 在计算目标用户的最近邻居集时, 只是根据用户项目评分矩阵来计算, 并没有考虑用户偏好和用户评分与项目之间的偏好关系, 以及在预测阶段没有对预测值进行判定处理.

用户偏好是指用户对评分的一个评分趋势, 通常会有两种情况: 积极评价(两个用户对项目都趋于高分评价); 消极评价(两个用户对项目都趋于低分评价). 传统的推荐算法在计算用户的邻居集时并没有考虑用户偏好, 只是所有的评分进行统一处理最终得到一个综合相似度, 根据该值的高低, 来判断用户之间是否相似, 这样的推荐处理可能会把用户讨厌的项目推荐给相似用户, 与实际情况不符, 例如甲喜欢项目 A 和 B, 对它们的评分都是 5, 用户乙讨厌项目 A 和 B, 对它们的评分都是 1, 可是在使用相似性方法计算最近邻居集时, 他们的非常相似, 这样就可能造成把一个用户讨厌的项目当成是另一个用户喜好的项目进行推荐. 通常在推荐系统中都只是采用用户之间的喜欢相似进行推荐, 这样会导致推荐不准, 从而影响推荐质量.

项目属性是指一个项目可能只属于一个类型也有可能同时属于多个属性类型, 例如一个用户对某一个电影进行了高分评价, 可是该电影同时属于喜剧和恐怖, 那么不考虑这些属性之间的关系而直接根据用户评分来进行推荐根本不能知道用户喜欢这个电影的真正兴趣, 到底是因为喜剧而喜欢或者是因为恐怖而喜欢或者是两个因素都有呢! 这样同样会导致最终的推

荐结果不准.

由于使用 UBCF 得到的预测值通常都是小数, 然而在实际推荐中, 用户对项目的推荐都是等级评分, 也就是一系列能够反映用户偏好的整数值, 这样同样会影响推荐的精度, 因此预测值判定也很重要, 预测值判定就是表示对预测值进行取整处理, 目前大多算法都是采用了简单的"四舍五入"的方法[16], 该方法虽然能够达到取整的效果, 却不能体现出用户真正的评分趋势. 李永[17]等提出了一种新的预测值判定方法, 通过趋势度、偏离度和判定度. 这三者的关系来定夺预测值的取整情况, 在性能上比"四舍五入"方法好, 不过要分别计算出前面三个变量, 算法复杂度很高, 效率较低. 因此本算法将针对上面所存在的问题进行改进, 以提高算法的推荐精度.

## 2　基于用户偏好和项目属性的协同过滤推荐算法(UPPPCF)

### 2.1 基于用户偏好的用户项目属性评分矩阵

基于上述分析, 针对 UBCF 算法存在的不足提出了以下解决方案: 为了不加大用户的反馈工作量和能够共享传统推荐系统中的用户信息数据, 需要综合考虑用户的偏好、用户评分与项目属性之间的关系来创建一个基于用户偏好和项目属性的用户项目属性评分矩阵, 由于这两个矩阵在结构上是一致的, 因此可以借助传统推荐算法中的相似性计算方法以及推荐方法来完成整个推荐.

由于用户偏好能够更准确的把用户喜好的兴趣聚集起来, 同时避免把一个用户讨厌的项目当成是另一个用户喜好的项目进行推荐, 在考虑用户偏好的基础上再计算用户项目评分与项目属性之间的关系, 能够发现用户积极评分下对某一个项目喜爱的真正原因, 也就是说能够找出用户真正喜爱的是该项目的某一属性, 因此结合用户偏好和项目属性能够更加明确的发现用户的兴趣偏好, 以此来查找的用户邻居集会更加的准确. 在本文中把用户偏好定义在一个特定的积极评分区域, 因此在确定了用户偏好以后再进行项目属性与用户项目评分矩阵之间的关系不会重复的进行计算, 并且由于项目属性是固定不变的, 在一段时间内用户的评分喜好相对稳定, 因此求解基于用户偏好的用户项目属性评分矩阵可以离线进行, 可以提高推荐的实时性.

基于用户偏好这个因素, 在这里先定义一个区域, 该区域主要体现用户对项目的积极评分, 然后根据该区域的评分求解邻居集, 就不会造成把一个用户讨厌的项目当成是另一个用户喜好的商项目进行推荐, 下面给出用户积极评价的定义:

定义 1. 目标用户 a 评分的项目集合是 $I_a=\{\mathrm{i}\,|\,R_{a,i}>=1\}$, $|I_a|$ 表示是目标用户 a 评过分的项目数量, 用户 a 的积极评价项目集则为 $I_a^{top}=\{\mathrm{i}\,|\,R_{a,i}>=3\}$, $I_a^{top}$ 则表示用户 a 的积极评分项目数量, 其中 $|I_a^{top}|\subseteq|I_a|$.

项目的属性有(不知道、动作、冒险、动画、儿童类、喜剧、犯罪、记录、剧情、奇幻、黑色、恐怖、音乐剧、神秘、爱情、科幻、惊悚、战争、西部), 如果该项目具有某一个或者多个属性时, 对应的项目属性值则为 1, 否则为 0. 综合考虑了用户偏好、用户项目评分矩阵以及项目属性矩阵创建一个基于用户偏好的用户项目属性评分矩阵, 如下表 1, 其中对应的值则表示用户对某一个项目的某一属性的评分.

表 1　基于用户偏好的用户项目属性评分矩阵

| 用户 | 项目 1 | 项目 2 | … | 项目 n |
|---|---|---|---|---|
| | $I_{1,1}\ldots I_{1,k}$ | $I_{2,1}\ldots I_{2,k}$ | … | $I_{n,1}\ldots I_{n,k}$ |
| $U_1$ | $I_{1,1,1}\ldots I_{1,1,k}$ | $I_{1,2,1}\ldots I_{1,2,k}$ | … | $I_{1,n,1}\ldots I_{1,n,k}$ |
| $U_2$ | $I_{2,1,1}\ldots I_{2,1,k}$ | $I_{2,2,1}\ldots I_{2,2,k}$ | … | $I_{2,n,1}\ldots I_{2,n,k}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $U_m$ | $I_{m,1,1}\ldots I_{m,1,k}$ | $I_{m,2,1}\ldots I_{m,2,k}$ | … | $I_{m,n,1}\ldots I_{m,n,k}$ |

基于上述的分析, 要得到基于用户偏好的用户项目属性评分矩阵, 必须了解用户的真正偏好, 首先求出用户的积极评价的项目集, 在此基础上查找用户对项目的各个属性的偏好程度, 最后结合用户项目评分矩阵和项目属性矩阵来进行求解, 详细的方法见算法 1.

算法 1. 计算基于用户偏好的用户项目属性评分矩阵

输入: 用户项目评分矩阵、项目属性评分矩阵

输出: 用户项目属性标签评分矩阵

步骤:

1) 根据定义 1 来查找各个用户的积极评价项目集合

2) 依次扫描步骤 1 中所求得的集合, 然后对集合中的项目具有的属性进行统计, 同时对其进行排序, 找出该用户最为感兴趣的 5 个属性.

3) 结合步骤 2 中求得的项目的各个属性值总和、用户项目评分矩阵以及项目属性矩阵来对计算基于用

户偏好的项目属性评分矩阵中用户对某一个项目的某一属性的评分值, 如果该项目只具有一个属性时, 则把对应的用户评分矩阵中的评分值赋给该属性, 如果该项目同时具有多个属性时, 判断它所具有的属性是否属于步骤 2 中提高的项目集的前 5 属性, 如果不属于则判定为用户的兴趣与该属性无关, 将其忽略, 如果属于, 则按照公式(5)计算.

$$Rate_a = \frac{Count(\text{a})}{Count(\text{Item})} \times Rate \tag{5}$$

其中 $Count(\text{a})$ 为 步骤 2 中计算的积极评价项目集中属性 a 的总和, $Count(Item)$ 表示的是项目 Item 在积极评价项目集中所具有的属性的总和, Rate 表示在用户项目评分矩阵中的对应评分.

4) 依次对所有的用户重复执行步骤 1 到 步骤 3, 最终得到一个基于用户偏好的用户项目属性评分矩阵.

本文主要采用修正过的余弦相似性方法来计算目标用户的邻居集. 在求解邻居集时依据前面所创建的基于用户偏好的用项目属性评分矩阵. 则它的相似性计算如公式(6)

$$sim(a,b) = \frac{\sum\limits_{i\in I_{ab}^{top}}\sum\limits_{j=1}^{S}(V_{a,i,j}-\bar{V}_a)(V_{\text{b},i,j}-\bar{V}_b)}{\sqrt{\sum\limits_{i\in I_a^{top}}\sum\limits_{j=1}^{S}(V_{a,i,j}-\bar{V}_a)^2}\sqrt{\sum\limits_{i\in I_b^{top}}\sum\limits_{j=1}^{S}(V_{b,i,j}-\bar{V}_b)^2}} \tag{6}$$

其中 S 表示的是项目属性标签为 1 的数量, $I_{ab}^{top}$ 表示目标用户 a 与用户 b 所共同表现出积极评分态度的项目集合 $I_a^{top}$, $I_b^{top}$ 分别表示用户 a 和 b 的积极评价项目集合. $V_{a,i,j}$ 表示的是用户 a 对项目 i 的第 j 个属性的评分值, 剩下的符号的意义和传统的相似性度量方式的意义相同.

### 2.2 预测评分值判定

在 UBCF 算法中还有一个至关重要的一部就是通过预测评分来产生输出接口, 一旦计算出当前用户的邻居集合时, 接下来就是采用相对应的方法预测目标用户的评分, 所采用的方法是前面的公式(4), 由于预测值判定取决于最初用户项目评分矩阵中的每个用户的实际评分, 由于采用实验数据集一共有 5 个评分等级, 分别对应 1、2、3、4、5, 所以, 有必要对预测的评分值进行判定, 使之与评分等级相对应. 因此在这里我采用了一种在比赛采用的积分方式, 也就是把评分分为最高分、最低分和正常评分三部分, 分别表示为高分评分区域, 低分评分区域, 正常评分区域, 本文的预测值判定的规则如下:

对任意一个用户 u, 求出该用户评分中等于 5 的项目个数总数用 $\left|I_a^{rate=5}\right|$ 表示, 这部分作为高分评分区域; 把评分从 2 到 4 的项目为通常评分区域, 这部分的项目个数总数表示为 $\left|I_a^{2\le rate\le 4}\right|$; 把评分等于 1 的项目作为低分评分区域, 这部分的项目个数总数表示为 $\left|I_a^{rate=1}\right|$. 然后分别比较他们之间的个数总数, 对任意的项目 $i$, $i\in Neiber_u$, 如果是处于高分评分区域, 则为 $P_{u,i}=\lceil P_{u,i}\rceil$, 如果是处于低分评分区域, 则为 $P_{u,i}=\lfloor P_{u,i}\rfloor$, 如果是处于通常评分区域, 则为 $P_{u,i}=Math.round(P_{u,i})$ 也就是对其使用四舍五入的方法进行取整.

使用该方法来进行预测值判定, 能够降低“偶然评分”对用户的真实的评价趋势的影响, 从而能够得到更准确的用户评分趋势, 从而使经过预测值判定后的值更接近真实情况, 从而提高推荐精度.

### 2.3 算法的主要步骤

基于以上的分析, 本文 UPPPCF 算法和传统的 UBCF 算法的操作过程基本是相同的, 不过在求解目标用户的邻居集时采用了新的基于用户偏好的用户项目属性评分矩阵, 同时分析用户的评分趋势来进行预测值判定. 具体的算法过程如下:

算法名称: 基于用户偏好和项目属性的协同过滤推荐算法(UPPPCF)

输入: 目标用户 a,基于用户偏好的用户项目属性评分矩阵

输出: 目标用户 a 的 N 个推荐项目

方法:

1) 寻找目标用户与其他用户共同评分的项目集

2) 利用基于用户偏好和项目属性标签的用户项目属性评分矩阵, 使用公式(6)对其进行相似性计算.

3) 将与目标用户 a 相似度最高的前 k 用户作为其邻居用户集 $UN_a$.

4) 利用公式(4)综合邻居用户的评分并预测用户 a 对 j 的预测评分.

5) 对其预测评分按照 2.2 节所描述的方法进行预测值判定.

6) 将预测评分最高的前 N 项目作为推荐项目.

本文 UPPPCF 与传统的协同过滤推荐算法, 以及前面所述的算法最大的区别在于计算目标用户相似性时所依据的矩阵不一样, 传统的协同过滤推荐算法在计算用户相似性时通常只考虑用户项目评分矩阵的评

分，并没有全面分析用户偏好以及项目属性对该评分所造成的影响，这样进行相似性计算比较片面同时推荐精度不高，针对这一问题，文件综述中提到了多种解决方法. 虽然上述方法都能在一定程度上提高推荐精度，但是都只是在计算用户相似性时将用户行为或者项目属性线性结合起来，并没有综合用户偏好、用户项目评分矩阵以及项目属性之间的真正关系，不能准确的表示用户的真正兴趣.

因此如果要提高算法的推荐进度，就必须先了解用户偏好、用户项目评分以及项目属性之间的真正关系. 因此本文算法综合考虑用户偏好、项目属性标签，建立一个更符合用户兴趣依据的用户项目属性评分矩阵，该矩阵能够更加直观的体现用户的兴趣偏好所在，然后再计算相似性，从而能够更加精确地找到目标用户的邻居集.

## 3　实验评估

### 3.1　数据集选取和度量标准

本文使用 MovieLens 数据集，选取其中公开的 ml-100k 数据集，该数据集包含了 943 个用户在 1682 部电影上的 100000 条评分记录，其中，每个用户至少对 20 部电影进行了评分，该分值的范围是从 1~5 的整数 ，不同的评分表达了每个用户的不同需求和兴趣，从 1 到 5 之间，分值越高表示用户对该电影越喜爱. 同时项目属性包括(不知道、动作、冒险、动画、儿童类、喜剧、犯罪、记录、剧情、奇幻、黑色、恐怖、音乐剧、神秘、爱情、科幻、惊悚、战争、西部的)19 种情况，该实验数据的稀疏度为 $1-\frac{100000}{943\times1682}=0.9639$，本文采用的度量标准是最常使用的平均绝对误差(MAE)，通过比较预测值和用户实际的评分值之间的偏差来衡量预测结果的准确性，MAE 越小，准确性越高. 目标用户的预测评价集合为 $P_u=\left(P_{u,i}\middle|i=1,2,\cdots,n\right)$，对应的目标用户实际评分集合为 $R_u=\left(r_{u,i}\middle|i=1,2,\cdots,n\right)$. 对于每个不为零的预测评价对 $\left\langle p_{u,i},r_{u,i}\right\rangle$ 按照公式(7)进行计算.

$$MAE=\frac{\sum_{i\in I_u}\mathrm{p}_{u,i}-r_{u,i}}{|I_u|} \tag{7}$$

其中，$I_u$ 为测试集内目标用户 u 的预测值和真实评价值都不为 0 的项目集合.

### 3.2　实验结果分析

为了检验算法的有效性，把本文提出的 UPPPCF(基于用户偏好和项目属性)算法与传统的 UBCF 算法，文献[11] (基于用户行为)，文献[12](基于用户兴趣点和特征)以及文献[14](基于项目相关性)推荐算法进行比较，与此同时将 MAE 作为度量标准，采用了 80%的数据作为训练集，20%作为测试集. 同时实验中，采取目标用户的最近邻居个数从 5 增加到 40，间隔为 5，来查看不同的最近邻居集的大小对预测准确度的影响，实验结果如表 2.

表 2　UPPPCF 算法和其他算法的 MAE 值对比

| 邻居集 | UBCF | 文献[11] | 文献[12] | 文献[14] | UPPPCF |
|---|---|---|---|---|---|
| 5 | 0.9078 | 0.8412 | 0.7894 | 0.7533 | 0.7671 |
| 10 | 0.8836 | 0.8173 | 0.7635 | 0.7284 | 0.7213 |
| 15 | 0.8754 | 0.8412 | 0.7546 | 0.7176 | 0.7013 |
| 20 | 0.8671 | 0.8015 | 0.7356 | 0.7267 | 0.6924 |
| 25 | 0.8572 | 0.7841 | 0.7278 | 0.7034 | 0.6720 |
| 30 | 0.8469 | 0.7561 | 0.6895 | 0.6853 | 0.6518 |
| 35 | 0.8376 | 0.7016 | 0.6736 | 0.6335 | 0.6151 |
| 40 | 0.8254 | 0.6814 | 0.6653 | 0.6072 | 0.5936 |

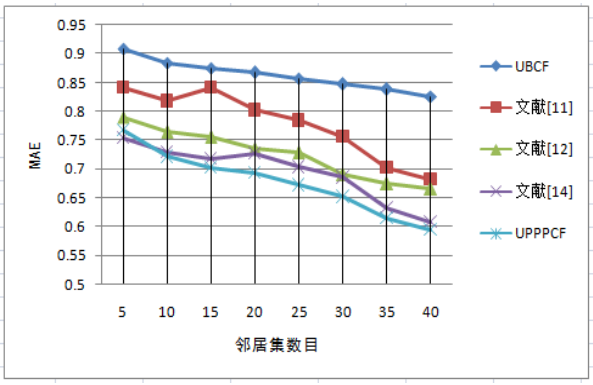


图 1　推荐算法的 MAE 比较

从图 1 可以看出，本文 UPPPCF 算法的推荐精度比传统的 UBCF、文献[11] 、文献[12]以及文献[14]都要高，其原因在于本算法综合考虑用户偏好、项目属性标签，建立一个更符合用户兴趣依据的用户项目属性评分矩阵，然后再计算相似性，从而能够更加精确地找到目标用户的邻居集. 同时对所求的预测值进行基于用户评分趋势影响的判定方法，在判定的过程中，剔除“偶然评分”对用户评分趋势的影响，提高了算法的推荐精度. 实验结果进一步表明：当目标用户的邻居集越多，能反映用户兴趣变化的用户越多，推荐的精度也越高.

## 4 结论

本文针对传统协同过滤推荐算法存在的不足，提出了一种基于用户偏好和项目属性的协同过滤推荐算法 UPPPCF. 该算法综合考虑了用户偏好、项目属性与用户项目评分之间的关系，将用户项目评分矩阵转换成更能突出用户真正的兴趣偏好的用户项目属性评分矩阵. 同时本文提出了一种新的预测值判定方法，该方法对用户的偏好进行区分，剔除“偶然评分”对用户评分趋势的影响，最终得到一个更为准确的评分趋势，然后根据它来进行预测值判定. 实验结果证明 UPPPCF 算法能够有效地解决传统的推荐算法中计算相似性以及预测值判定阶段所存在的问题，提高了算法的推荐质量.

## 参考文献


1 Eppler MJ, Mengis J. The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. The Information Society, 2004, 20(5): 325–344.

2 Breese J, Heckeman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Proc. of the 14th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1998. 43–52.

3 Goldberg D, Nichols D, Oki B, et al. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 1992, 35(12): 61–70.

4 Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of Netnews. Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work. ACM. 1994. 175–186.

5 Shardanand U, Maes P. Social information filtering: Algorithms for automating “word of mouth”. Proc. of the SIGCHI Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co. 1995. 210–217.

6 黄创光,印鉴,汪静等.不确定近邻的协同过滤推荐算法.计算机学报,2010,33(8):1369–1376.

7 陈健,印鉴.基于影响集的协同过滤推荐算法.软件学报,2007,18(7):1685–1694.

8 Rajasangari RA. An author recommender system using both content-based and collaborative filtering methods. California State University, 2011.

9 吴湖,王永吉,王哲.两阶段联合聚类协同过滤算法.软件学报,2010,21(5):1042–1054.

10 陶俊,张宁.基于用户兴趣分类的协同过滤推荐算法.计算机系统应用,2011,20(5):55–59.

11 王义,马尚才.基于用户行为的个性化推荐系统的设计与应用.计算机系统应用,2010,19(8):29–33.

12 严冬梅,鲁城华.基于用户兴趣点和特征的优化协同过滤推荐.计算机应用研究,2012,29(2):497–500.

13 嵇晓声,刘宴兵,罗来明.协同过滤中基于用户兴趣的相似性度量方法.计算机应用,2010,30(10):2610–2618.

14 刘芳先,宋顺林,等.改进的协同过滤推荐算法.计算机工程与应用, 2011,47(8):72–75.

15 Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithm. Proc. of the 10th International Conference on World Wide Web Conference. New York. ACM Press. 2001. 285–295.

16 李春,朱珍民,高晓芳.基于邻居决策的协同过滤推荐算法.计算机工程,2010,36(13): 34–36,39.

17 李永,徐智国,张勇等.协作过滤算法中一种预测值判定方法的研究.小型微型计算机系统,2008,3(3):469–472.