

# 融合信任用户的协同过滤推荐算法<sup>①</sup>

林建辉, 严宣辉, 黄波

(福建师范大学 数学与计算机科学学院, 福州 350007)

**摘要:** 推荐系统中普遍存在的数据稀疏性问题使得协同过滤算法所要求的近邻搜索准确性降低, 以及搜索到的最近邻用户过少, 这对整个推荐系统的推荐质量和推荐的准确性产生重要影响, 而这个问题对于传统的协同过滤推荐是难以解决的. 针对这个问题, 通过将用户之间的信任关系与对项目的评分相似性相融合, 提出一种融合信任用户的协同过滤推荐算法, 利用有向网络图构建的用户之间的信任关系, 弥补了仅仅依靠计算用户间相似性不能准确衡量用户之间关系的缺陷. 实验结果证明, 该算法能够提高系统的推荐质量和准确性.

**关键词:** 推荐系统; 协同过滤; 有向网络; 信任关系; 数据稀疏性

## Collaborative Filtering Recommendation Algorithm Based on Trust Users

LIN Jian-Hui, YAN Xuan-Hui, HUANG Bo

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

**Abstract:** The common data sparsity in recommendation systems makes the nearest neighbor search is not accurate and lets the search results of the nearest neighbor is too small. This will affect the recommended quality and accuracy of the recommendation system, moreover it is difficult to solve in the traditional collaborative filtering recommendation. To overcome the difficulty of data sparsity in recommendation systems, a novel collaborative filtering algorithm is presented which is based on the combination of trust relationship between users and the similarity of scores of the projects. This algorithm constructs the trust relationship among users by using a directed network graph, which can make up the defect that the user's relationship cannot be accurately measured by the user's similarity. The experimental results show that the proposed algorithm can improve the quality and accuracy of the recommendation system.

**Key words:** recommendation system; collaborative filtering; directed network; trusting relationship; data sparsity

## 1 引言

信息过载<sup>[1,2]</sup>问题对于互联网络时代来说无疑是一种巨大的时间和资源上的浪费, 而个性化推荐技术的产生恰到好处的解决了这个问题. 一个完整的个性化推荐系统<sup>[3]</sup>包括用户建模模块、推荐对象建模模块以及推荐算法模块, 其中推荐算法模块是其最为核心的部分. 协同过滤推荐算法(collaborative filtering recommendation algorithm, CF)<sup>[4]</sup>不需要分析用户、项目等的特征属性, 对推荐对象没有特殊的要求, 能够处理非结构化的复杂对象, 因此在现实中得到更多研究者的关注. 协同过滤算法包括基于领域(neighborhood-based)<sup>[5,6]</sup>和基于模型(model-based)<sup>[7]</sup>的

两类, 而基于领域的推荐算法又可分为基于用户(user-based)<sup>[5]</sup>和基于项目(item-based)<sup>[6]</sup>两种. User-based 的推荐算法是根据用户之间的相似爱好, 计算其相似度, 为目标用户选取兴趣相似的邻居, 并由此为目标用户推荐邻居喜欢的项目. Item-based 的推荐算法则是根据用户喜欢项目的相似项目, 计算其相似度, 为目标用户选取其喜爱项目相似度最大的前 N 个项目, 作为推荐结果. Model-based 的推荐算法是根据有监督的学习方法得到推荐模型, 利用该模型产生推荐结果. User-based 的推荐算法在现实生活当中更能为用户推荐新颖项目, 而且对产生的推荐结果也能够做出合理的解释, 因此得到更为广泛的应用.

① 收稿时间:2016-09-18;收到修改稿时间:2016-11-14 [doi:10.15888/j.cnki.csa.005805]

然而, User-based 的推荐算法存在这样的问题: 首先, 传统的相似性度量仅仅考虑了用户对项目的评分, 这其实很难得到准确度较高的推荐结果. 其次, 对于数据的稀疏性问题<sup>[8,9]</sup>也没有较好的应对策略. 对此, 研究者提出了很多不同的解决方案. 文献[10]的方法是先对由初始数据得到的评分矩阵进行降维, 在此基础上计算用户的相似性, 并且采用奇异值分解的技术填补邻居用户的缺失值, 以降低邻居用户的评分稀疏性. 文献[11]采用传统的 K-means 聚类方法, 对项目进行聚类, 计算根据聚类得到的聚类中心与目标项目的相似性, 最后从相似性最大的类中找寻邻居用户并产生推荐. 文献[12]提出了加加权二部图推荐算法, 从资源分配的角度建立项目-用户关系图, 利用项目的评分作为二部图中用户与项目的边权, 按照用户-项目间边权占该节点权重和的比例分配资源, 从而实现评分值高的项目得到优先推荐. 文献[13]的方法是收集的用户的历史行为数据, 并利用这些数据构建一个用户间的信任网络, 计算用户与目标用户的评分相似性与信任值, 由此基础产生邻居用户, 进而进行预测推荐. 文献[14]提出了融合评分和信任的用户偏好模型, 依据评分数据集设定共同评分资源数目的阈值, 有策略地选择用户间评分相似度和信任值, 计算用户偏好程度. 其次, 通过引入用户对资源情感内涵的偏好度量机制, 弥补了以往协同过滤推荐对用户情感的忽略.

这些研究人员的方法在一定程度上都取得了较好的推荐效果, 但同时也存在着各方面的问题: 空值填补本来就是对未知值的一种预测, 那么再由预测产生的预测结果准确性就有待考证. 过度的降维则会导致原始信息的损失, 所以矩阵分解也是一种比较难于保证取得良好效果的技术. 聚类技术受数据稀疏性的影响, 造成聚类质量不高, 产生的邻居用户因此也不够准确. 因此这些方法对解决数据的稀疏性仍然不够恰当. 另外, 在考虑到每个用户的社会关系, 添加信任机制的协同过滤相关文章中, 仍然存在涉及到用户的历史行为数据难以获取、关系到用户隐私安全保密等问题.

针对上述未能恰当解决数据稀疏性及仅考虑用户对项目的评分对预测结果不够准确的问题, 本文由用户间的信任关系和对项目的评分相似性提出一种融合信任用户的协同过滤推荐算法(collaborative filtering recommendation algorithm based on trust users, TUCF).

根据用户对项目的评分计算用户之间的相似性, 通过用户之间对项目的操作构造出用户-项目、项目-用户的有向网络图, 并由此构造出用户-用户的有向信任网络图, 发现一种新的用户之间对项目操作的简单信任关系. 通过综合考虑用户之间的相似性和信任关系选择最近邻居, 弥补了传统的相似性度量仅仅考虑了用户对项目评分的缺点, 有效地提升了推荐系统的质量及准确性并缓解了数据稀疏性所带来的问题.

本文接下来的工作: 第 2 节给出问题的定义, 介绍协同过滤推荐算法的基本方法; 第 3 节详细介绍本文提出的融合信任用户的协同过滤推荐算法; 第 4 节针对本文算法进行实验验证和分析; 最后是对本文的总结.

## 2 问题定义与基本方法

在协同过滤推荐系统中,  $m$  个用户用集合  $U=\{u_1, u_2, \dots, u_m\}$  表示,  $n$  个项目用集合  $I=\{i_1, i_2, \dots, i_n\}$  表示, 用户对项目的评分数据集用一个  $m \times n$  阶的矩阵  $R$  表示, 其形式如下:

$$R = \begin{bmatrix} R_{1,1} & \dots & R_{1,j} & \dots & R_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{i,1} & \dots & R_{i,j} & \dots & R_{i,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{m,1} & \dots & R_{m,j} & \dots & R_{m,n} \end{bmatrix}$$

其中,  $R_{i,j}(1 \leq i \leq m, 1 \leq j \leq n)$  表示用户  $u_i$  对项目  $i_j$  的评分. 评分值范围为 0—5, 0 分表示用户未对该项目进行评价, 评分值越大表示用户对该项目越喜欢. 要强调的是该矩阵通常是一个稀疏矩阵, 而推荐系统的工作就是根据现有的数据集对未知评分的预测.

### 2.1 用户相似性度量

用户之间相似性的度量方法通常有三种<sup>[15]</sup>: 标准的余弦相似性, Pearson 相关系数(Pearson Correlation Coefficient)相似性和修正的余弦相似性. 本文以 Pearson 相关系数相似性的度量方法为基础, 公式如式(1)所示:

$$\text{sim}(u_a, u_b) = \frac{\sum_{i_k \in I_a \cap I_b} (R_{a,k} - \bar{R}_a) \times (R_{b,k} - \bar{R}_b)}{\sqrt{\sum_{i_k \in I_a \cap I_b} (R_{a,k} - \bar{R}_a)^2} \sqrt{\sum_{i_k \in I_a \cap I_b} (R_{b,k} - \bar{R}_b)^2}} \quad (1)$$

其中,  $\text{sim}(u_a, u_b)$  表示的用户  $u_a$  与  $u_b$  的相似度, 其值在 -1 和 +1 之间, 该值越大, 说明用户  $u_a$  与  $u_b$  就越相似.  $R_{a,k}$  和  $R_{b,k}$  分别为用户  $u_a$  和  $u_b$  对项目  $i_k$  的评分,  $\bar{R}_a$  和  $\bar{R}_b$  分别表示用户  $u_a$  和  $u_b$  的对项目的平均评分,  $I_a \cap I_b$

是用户  $u_a$  与  $u_b$  共同评分的项目集合.

### 2.2 预测推荐

对目标用户  $u_a$  的未评分项目进行预测, 首先需要找到对未评分项目有过评分的用户集合  $S$ , 公式(1)计算  $u_a$  与集合  $S$  中用户的相似性大小, 取相似性最大的  $K$  个用户作为  $u_a$  的邻居用户集  $S(u_a)$ . 根据公式(2)<sup>[17]</sup> 计算目标用户  $u_a$  对未评分项目的预测评分.

$$P_{a,j} = \bar{R}_a + \frac{\sum_{u_b \in S(u_a)} (R_{b,j} - \bar{R}_b) \times sim(u_a, u_b)}{\sum_{u_b \in S(u_a)} sim(u_a, u_b)} \quad (2)$$

式中,  $P_{a,j}$  为目标用户  $u_a$  对未知评分项目  $i_j$  的预测评分,  $\bar{R}_a$  和  $\bar{R}_b$  分别为目标用户  $u_a$  和邻居用户  $u_b$  对项目的平均评分,  $R_{b,j}$  表示用户  $u_b$  对项目  $i_j$  的评分,  $sim(u_a, u_b)$  表示的用户  $u_a$  与  $u_b$  的相似度. 最终将评分最高的前  $N$  个项目作为目标用户  $u_a$  的推荐结果.

### 3 融合信任用户的协同过滤推荐算法

传统的基于用户的协同过滤推荐算法(user-based collaborative filtering recommendation algorithm, UCF)在度量用户之间的相似性时仅仅考虑了用户对项目的评分, 这其实很难得到准确度较高的推荐结果. 另外, 对于数据集稀疏性的情况, 这种度量方式就显得更加不够合理. 在日常生活当中以及电子商务领域, 用户之间的信任关系表现出日益重要的作用. 许多学科学者给出了对信任的不同定义: Gambetta(2000)<sup>[16]</sup>将信任定义为主体为谋取福利而表现出的对客体执行某些工作的依赖. 这个定义显示了信任关系中主、客体之间的依赖性和可靠性的概念. Josang(2007)<sup>[17]</sup>认为信任是主体在特定情况下对某物或某人表现出相对安全感的程度, 尽管可能产生负面的后果. 相关文献[13]和文献[14]通过用户的历史行为数据构建用户间信任关系, 但存在着涉及历史行为数据难以获取以及用户的隐私保密安全问题. 我们利用用户之间对项目的评分数据构建用户-项目的有向图及项目-用户的有向图<sup>[12]</sup>, 最终得到用户-用户的有向图, 计算得到用户之间的交互信任关系的权重, 结合用户之间的评分相似性与信任关系来度量用户之间的关系, 以此来弥补传统的基于用户的协同过滤算法的不足.

#### 3.1 用户信任关系的构建

本文将用户与项目之间的关系转化为一种选择关

系, 并用有向网络图  $G<U, I>$  表示, 其中  $U$  代表的是用户节点,  $I$  代表的是项目节点. 用有向边  $e<u_i, i_j>$  表示用户  $u_i$  对项目  $i_j$  有操作, 并用数字 1 表示, 而如果用户对项目没有操作则不存在边关系, 用数字 0 表示, 其边构造如式(3):

$$e_{u_i, i_j} = \begin{cases} 1 & R_{i,j} \neq 0 \\ 0 & R_{i,j} = 0 \end{cases} \quad (3)$$

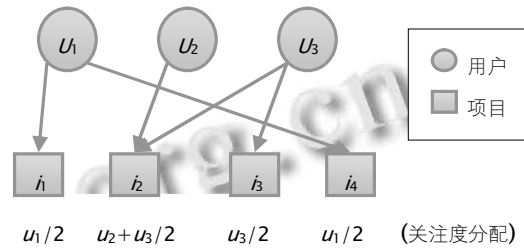


图1 用户-项目的有向网络图

在构建用户-项目的有向网络图过程中, 假设认为每个用户对有选择操作的项目的关注度是相同的, 即将用户的关注度平均分配给每一个用户有过操作的项目. 用户-项目的有向网络图如图 1 所示, 图中用户对项目节点的关注度计算公式如(4)所示:

$$a(i_j) = \sum_{i=1}^m \frac{e_{u_i, i_j}}{r(u_i)} u_i \quad (4)$$

其中,  $r(u_i)$  为用户  $u_i$  关注的项目总数,  $e_{u_i, i_j}$  表示用户  $u_i$  对项目  $i_j$  是否有关注操作,  $a(i_j)$  表示项目  $i_j$  受到用户的关注度,  $j=1, 2, \dots, n$ .

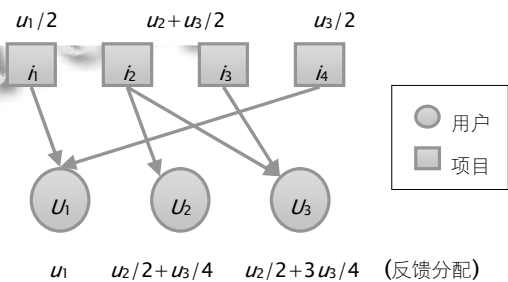


图2 项目-用户的有向网络图

构建项目-用户有向网络图过程类似于用户-项目的有向网络图构建, 其构造如图 2 所示, 这里将项目得到的用户的关注度重新反馈给用户, 反馈表达式为(5):

$$f(u_i) = \sum_{j=1}^n \frac{e_{i_j, u_i}}{r(i_j)} a(i_j) \quad (5)$$

式中,  $r(i_j)$  为项目  $i_j$  受到用户关注的总数,  $e_{i_j, u_i}$  表示项目  $i_j$  是否受到用户  $u_i$  的关注,  $f(u_i)$  表示项目反馈给用

户  $u_i$  的关注度,  $i=1, 2, \dots, m$ .

根据用户-项目、项目-用户的用向网络图得到最终的用户-用户的有向网络图, 如图 3 所示.

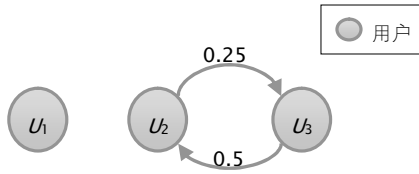


图 3 用户-用户的有向网络图

可以将公式(4)和(5)结合得到式(6):

$$f(u_i) = \sum_{j=1}^n \frac{e_{ij, u_i}}{r(i_j)} \sum_{i=1}^m \frac{e_{u_i, ij}}{r(u_i)} u_i \quad (6)$$

我们定义:

$$T_{u_a, u_b} = \frac{1}{r(u_b)} \sum_{j=1}^n \frac{e_{ij, u_a} \cdot e_{u_b, ij}}{r(i_j)} \quad (7)$$

公式(7)中, 我们用字母  $T_{u_a, u_b}$  来代表用户  $u_a$  对用户  $u_b$  在项目上的交互信任值. 其表示的是用户关系的一种交互信任程度, 取值范围为 0 到 1 之间, 0 表示两个用户没有交互操作过, 不存在信任关系, 信任值越大, 那么两个用户的信任程度也就越高.

因此式(6)可以简化为:

$$f(u_i) = \sum_{j=1}^m T_{u_i, u_j} \cdot u_j \quad (8)$$

公式(8)所表示的是用户  $u_i$  与其他用户之间的关系, 在本文我们将用户存在的这种关系理解为用户之间在项目上交互的信任关系. 在图 3 中, 用户  $u_2$  到  $u_3$  的有向边的权值表示用户  $u_2$  对  $u_3$  的信任度值, 用户  $u_3$  到  $u_2$  的有向边的权值表示用户  $u_3$  对  $u_2$  的信任度值, 因为用户  $u_1$  没有对用户  $u_2$ 、 $u_3$  关注操作的项目产生任何操作, 而用户  $u_2$ 、 $u_3$  对用户  $u_1$  关注的项目也没有兴趣, 所以用户  $u_1$  与用户  $u_2$ 、 $u_3$  之间没有信任值.

至此, 用以表示用户之间信任关系的有向网络图构建完成, 值得一提的是, 用户之间的信任关系还包括两个重要特性, 第一: 主观选择性, 不同的用户对同一个用户的信任值不同. 第二: 非对称性, 用户之间的信任是单向的, 即用户  $u_a$  对用户  $u_b$  的信任值不等于用户  $u_b$  对  $u_a$  的信任值.

### 3.2 邻居选择策略

传统的基于用户的协同过滤算法在邻居的选择过程中仅仅依赖用户之间的相似性大小, 相似性大则更有可能作为目标用户的邻居, 而用户之间的相似性度

量是通过用户之间的共同评分, 当用户之间的共同评分过少, 而评分又比较接近, 此时的相似性度量必然存在一定的偶然性, 选择的邻居用户就不够合理. 因此, 本文在邻居用户的选择方面做出了一些调整工作:

① 设置共同评分项目个数  $\eta$

$$I_a \cap I_b \geq \eta$$

② 相似度计算在①的基础上设置调节因子  $\theta$

$$sim(u_a, u_b) \geq \theta$$

③ 设置局部评分差异因子  $\Phi$  和个数  $\mu$

$$N(R_{a, j} - R_{b, j} \leq \Phi) \geq \mu$$

④ 信任度计算在③的基础上设置调节因子  $\xi$

$$T(u_a, u_b) \geq \xi$$

上述①和③出现的  $a$  和  $b$  指的是用户  $u_a$  和  $u_b$ ,  $j$  表示的是项目  $i_j$ , 最终根据这四个条件获得目标用户  $u_a$  的邻居用户集  $S(u_a)$ .

### 3.3 预测推荐

基于以上的研究, 对于目标用户  $u_a$  对未知评分的预测, 我们给出以下公式:

$$W(u_a, u_b) = \lambda \times sim(u_a, u_b) + (1 - \lambda) \times T(u_a, u_b) \quad (9)$$

$$P_{a, j} = \bar{R}_a + \frac{\sum_{u_b \in S(u_a)} (R_{b, j} - \bar{R}_b) \times W(u_a, u_b)}{\sum_{u_b \in S(u_a)} W(u_a, u_b)} \quad (10)$$

公式(9)中,  $W(u_a, u_b)$  表示用户  $u_a$  与  $u_b$  融合相似度与信任关系的权重,  $\lambda$  为相似度和信任关系的融合参数,  $sim(u_a, u_b)$  表示的用户  $u_a$  与  $u_b$  的相似度,  $T(u_a, u_b)$  表示用户  $u_a$  与  $u_b$  的信任关系值. 公式(10)中  $P_{a, j}$  为目标用户  $u_a$  对未知评分项目  $i_j$  的预测评分,  $\bar{R}_a$  和  $\bar{R}_b$  分别为目标用户  $u_a$  和邻居用户  $u_b$  对有评分项目的平均评分, 用户  $u_b$  属于目标用户  $u_a$  的邻居用户集  $S(u_a)$ ,  $R_{b, j}$  表示用户  $u_b$  对项目  $i_j$  的评分. 最终将预测评分最高的  $N$  个项目作为目标用户  $u_a$  的推荐结果.

### 3.4 算法描述

通过第 1 和第 2 节的研究讨论, 得到了下表对本文 TUCF 算法的描述:

算法: 一种融合信任用户的协同过滤推荐算法(TUCF)
输入: 用户对项目的评分数据集, 目标用户 $u_a$ , 未知评分项目 $i_j$ , 各个设置邻居用户因子和融合参数 $\lambda$ ;
输出: 用户 $u_a$ 对项目 $i_j$ 的预测评分 $P_{a, j}$ ;
Step1. 由用户评分数据集得到用户-项目矩阵 $R$ , 根据公式(1)计算用户 $u_a$ 与其他用户的相似度;

Step2. 构建用户  $u_a$  与其他用户之间的信任关系, 根据公式(8)

得到用户之间的信任权重, 标准化信任矩阵;

Step3. 找到对目标项目  $i_j$  有评分的邻居, 根据邻居选择策略找到用户  $u_a$  的邻居用户集  $S(u_a)$ ;

Step4. 选择适合的融合参数  $\lambda$ ;

Step5. 由预测公式(10)计算得到目标用户  $u_a$  对未知评分项目  $i_j$  的预测评分  $P_{a,j}$ .

在一个包含  $m$  个用户和  $n$  个项目数据集的实验当中, Step1 中计算用户相似度的时间复杂度为  $O(m \cdot n)$ ; 而在 Step2 中首先需要构建用户之间的信任关系, 其时间复杂度为  $O(m^2 \cdot n)$ ; Step5 中预测未知项目的评分需要的时间复杂度为  $O(m \cdot n)$ . Step1 和 Step2 两个步骤都可以离线进行, 减少在线计算的时间, 提高系统的推荐效率. 综合上述分析, 本文算法的时间复杂度为  $O(m^2 \cdot n)$ .

## 4 实验与结果分析

### 4.1 实验数据集

本文实验所采用的数据集包括: (1)美国明尼苏达大学 GroupLens 研究项目组提供的 MovieLens 数据集 ML-100K(<http://MovieLens.umn.edu/>), 数据集包含了 943 个用户对 1682 部电影共 100000 条的评分记录, 其数据的稀疏程度为  $1 - 100000 / (943 \times 1682) = 0.9370$ . (2) 百度 2013 举办的百度电影推荐系统比赛所用的数据集(<http://openresearch.baidu.com/ark.jspx>), 数据集包含了 15 万用户对 15000 部电影约 1 百万条的评分记录, 实验随机选择了 943 位用户对 1682 部电影共 62507 条的评分记录, 其数据的稀疏程度为  $1 - 62507 / (943 \times 1682) = 0.9606$ . 两个数据集的评分范围为 1~5, 用户喜欢该部电影的程度由值大小依次递增, 实验随机选取数据集集中的 80% 作为训练集, 20% 作为测试集.

### 4.2 评价准则

本文采用的评价准则是广泛应用于评价协同过滤推荐算法的平均绝对误差(mean absolute error, MAE)<sup>[18]</sup>和精确度(Precision)<sup>[19]</sup>, MAE 通过计算预测评分与实际评分的平均误差来衡量推荐质量, MAE 越小, 说明推荐质量越好. 其计算公式如(11)所示, 预测用户对项目的评分值为  $p_{u,i}$ , 而实际的用户对项目的评分值为  $r_{u,i}$ ,  $T$  为测试集的项目数量.

$$MAE = \frac{\sum_{i=1}^T |p_{u,i} - r_{u,i}|}{T} \quad (11)$$

Precision 指标则是通过计算预测评分与实际评分相等的数量占整个测试集的比率来衡量推荐的准确度, Precision 指标值越大, 说明推荐准确性越好. 其计算公式如(12)和(13)所示, 用  $N_i$  表示用户对项目  $i$  预测评分值与实际评分值的关系, 当其值等于 1 时说明预测用户对项目评分值与实际评分值相等, 否则为 0.

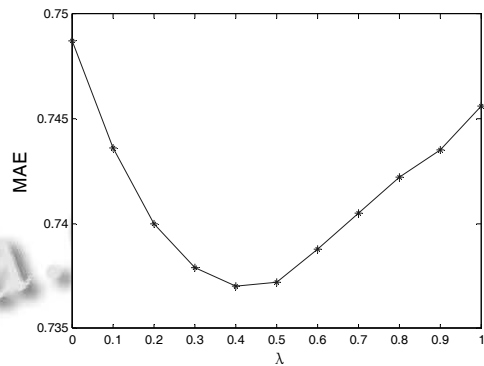
$$N_i = \begin{cases} 1 & p_{u,i} = r_{u,i} \\ 0 & p_{u,i} \neq r_{u,i} \end{cases} \quad (12)$$

$$Precision = \frac{\sum_{i=1}^T N_i}{T} \quad (13)$$

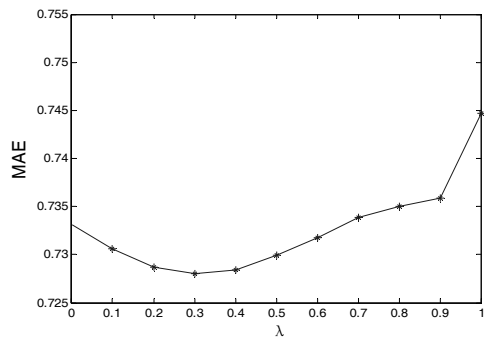
### 4.3 实验结果与分析

(1) 实验 1: 融合参数  $\lambda$  对实验结果的影响

这个实验的目的是为了选择最佳的融合参数  $\lambda$  使得推荐质量达到最好, 实验用横坐标表示融合参数  $\lambda$  的取值, 取值从 0 到 1, 每次增加 0.1. 纵坐标为 MAE, 通过观察 MAE 值的变化, 选取最恰当的融合参数, 实验中的我们任意选取邻居用户个数为 30. 实验结果如图 4 所示.



(a) MovieLens 数据集下融合参数  $\lambda$  对 MAE 的影响



(b) 百度电影数据集下融合参数  $\lambda$  对 MAE 的影响

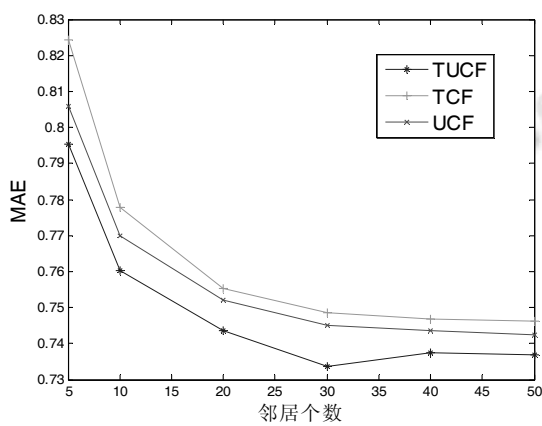
图 4 不同数据集下融合参数  $\lambda$  对 MAE 的影响

从图 4(a)、(b)中可以看出, 在不同数据集中随着融合参数  $\lambda$  的取值逐渐增大, MAE 值的变化大致可以看作为从大慢慢减小然后再逐渐增大. 在 MovieLens 数据集下, 当融合参数  $\lambda$  的取值为 0.4 时, MAE 达到最小, 此时的推荐质量达到最好, 因此后续 MovieLens 数据集下的实验融合参数  $\lambda$  的取值为 0.4. 而在百度电影数据集下, 当融合参数  $\lambda$  的取值为 0.3 时, MAE 取得最小值, 此时的推荐质量达到最好, 因此后续百度电影数据集下的实验融合参数  $\lambda$  的取值为 0.3. 实验中, 不同数据集的融合参数  $\lambda$  取值不同, 由于不同的数据集其特征结构不

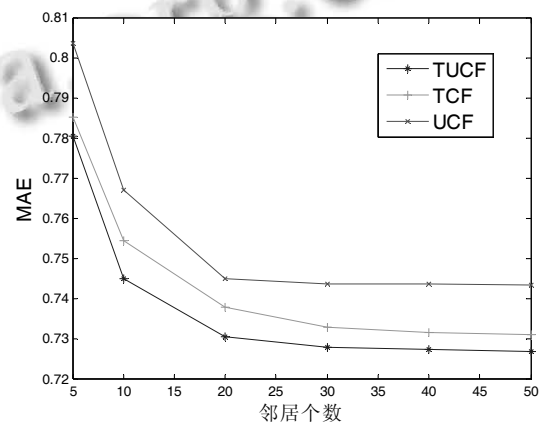
同, 稀疏程度也不同, 因此选取的融合参数不同. 实验能够根据各个数据集的稀疏程度选择最恰当的融合参数  $\lambda$ , 使得达到最佳的推荐质量.

(2) 实验 2: 与其它协同过滤推荐算法比较

为了验证本文融合信任用户的协同过滤推荐算法 (TUCF) 的推荐质量和准确性, 本文将与传统的基于用户的协同过滤推荐算法 (UCF)<sup>[3]</sup> 和本文仅依靠信任关系的协同过滤推荐算法 (TCF) 进行对比. 实验中横坐标为邻居的个数, 由 5 个逐渐增至 50 个邻居用户, 而纵坐标为 MAE 和 Precision 等指标. 实验结果如图 5、6 所示.

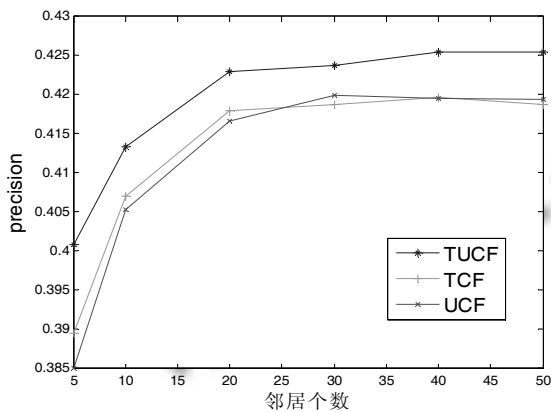


(a) MovieLens 数据集下各推荐算法的 MAE 比较

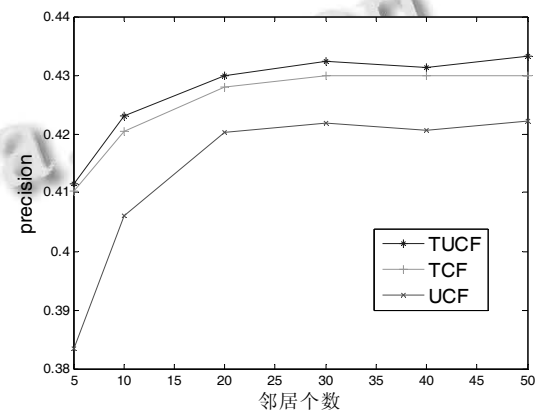


(b) 百度电影数据集下各推荐算法的 MAE 比较

图 5 不同数据集下各推荐算法推荐质量比较



(a) MovieLens 数据集下各推荐算法的 Precision 比较



(b) 百度电影数据集下各推荐算法的 Precision 比较

图 6 不同数据集下各推荐算法推荐准确性比较

从图 5(a)、(b)中可以发现, 随着邻居用户个数的增加各个推荐算法在不同数据集下的 MAE 值都在降低, 达到一定的值都开始趋于平缓, 从总体上看 TUCF 的推荐质量都优于 TCF 和 UCF, 而 TCF 的推荐质量在不同的数据集下与 UCF 相比较有不同的结果.

而根据图 6(a)、(b)的结果可以知道, 各个推荐算法在不同数据集下的 Precision 指标值随着邻居用户个数的增加而增加, 当邻居用户的个数达到一定的值时也将开始趋于平缓, 从各个推荐算法的 Precision 指标值显示, TUCF 的推荐准确性要比 TCF 与 UCF 的高, 而 TCF

的推荐准确性也优于 UCF。第一,由于本文提出的用户信任关系能够较为准确地衡量用户之间的关系,得到更为准确的近邻,缓解了数据稀疏性所带来的近邻搜索不准确的问题;第二,将用户之间的信任关系与用户间的评分相似度相结合可以弥补仅靠评分相似度导致推荐准确度低的缺陷,从而提高了推荐系统的推荐质量和推荐准确性。

## 5 结语

本文提出的一种融合信任用户的协同过滤推荐算法弥补了仅依靠计算用户间相似性因而不能准确衡量用户之间关系的缺陷,针对没有任何用户之间信任关系数据的情况,根据用户对项目的操作构造出用户之间的有向信任网络图,发现了一种新的用户之间对项目操作的简单信任关系。将现实生活中用户之间的相互信任关系融入到计算用户之间的关系当中,更能够准确地衡量出用户之间的关系、找到目标用户的最近邻用户,有效地缓解了数据稀疏性所带来的近邻搜索不准确以及搜索到的最近邻用户过少等问题。

### 参考文献

- 1 许海玲,吴潇,李晓东,等.互联网推荐系统比较研究.软件学报,2009,20(2):350-362.
- 2 Borchers A, Herlocker J, Konstan J, et al. Ganging up on information overload. *Computer*, 1998, 31(4): 106-108.
- 3 Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowledge and Data Engineering*, 2005, 17(6):734-749.
- 4 Herlocker JL, Konstan JA, Borchers A, et al. An algorithmic framework for performing collaborative filtering. *International Acm Sigir Conference on Research & Development in Information Retrieval*. 1999. 230-237.
- 5 Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*. 1998. 43-52.
- 6 Deshpande M, Karypis G. Item-based top-N recommendation algorithm. *ACM Trans. on information system*, 2004, 22(1): 143-177.
- 7 Kamishima T, Akaho S. Nantonac collaborative filtering: a model-based approach. *Proc. of the 4th ACM Conference on Recommender Systems*. New York. ACM. 2010. 273-276.
- 8 Goldberg K, Roeder T, Gupta D, et al. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 2001, 4 (2): 133-151.
- 9 李华,张宇,孙俊华.基于用户模糊聚类的协同过滤推荐研究. *计算机科学*,2012,39(12):83-86.
- 10 冷亚军,梁昌勇,陆青.基于近邻评分填补的协同过滤推荐算法. *计算机工程*,2012,38(21):56-58.
- 11 邓爱林,左子叶,朱扬勇.基于项目聚类的协同过滤推荐算法. *小型微型计算机系统*,2004,25(9):1665-1670.
- 12 张新猛,蒋盛益.基于加权二部图的个性化推荐算. *计算机应用*,2012,32(3):654-657.
- 13 杜永萍,黄亮,何明.融合信任计算的协同过滤推荐方法. *模式识别与人工智能*,2014,27(5):417-425.
- 14 Qin J, Zheng Q, Tian F, et al. Collaborative filtering algorithm integrating trust and preference of user's emotion. *Journal of Software*, 2013, 24 (2): 61-72.
- 15 Su XY, Khoshgoftaar T. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 2009 (4): 421-445.
- 16 Gambett D. Can we trust trust? in Gambetta, Diego(ed.). *Trust: Making and breaking cooperative relations*, electronic edition, Department of Sociology, University of Oxford, 2000, chapter 13, 213-237.
- 17 Josang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 2007, 43(2): 618-644.
- 18 Papagelis M, Plexousakis D. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Application for Artificial Intelligence*, 2005, 18(7): 781-789.
- 19 王鹏,王晶晶,俞能海.基于核方法的 user-based 协同过滤推荐算法. *计算机研究与发展*,2013,50(7):1441-1451.