

# 一种基于用户聚类的协同过滤推荐算法

李 涛<sup>1,2</sup>, 王建东<sup>1</sup>, 叶飞跃<sup>1</sup>, 冯新宇<sup>1</sup>, 张有东<sup>1</sup>

(1. 南京航空航天大学信息科学与技术学院, 江苏 南京 210016;

2. 南京信息工程大学电子与信息工程学院, 江苏 南京 210044)

**摘 要:** 为解决传统协同过滤算法在生成推荐时的速度瓶颈问题, 提出了一种基于用户聚类的协同过滤推荐算法。该算法将推荐过程分成了离线和在线两个部分。离线时, 算法对基本用户数据进行预处理, 并对基本用户聚类; 在线时, 算法利用已有的用户聚类寻找目标用户最近邻居, 并产生推荐。实验表明, 基于用户聚类的协同过滤推荐算法不仅加快了推荐生成速度, 而且提高了推荐质量。

**关键词:** 推荐算法; 协同过滤; 聚类; 平均绝对误差

**中图分类号:** TP391

**文献标志码:** A

## Collaborative filtering recommendation algorithm based on clustering basal users

LI Tao<sup>1,2</sup>, WANG Jian dong<sup>1</sup>, YE Fei yue<sup>1</sup>, FENG Xin yu<sup>1</sup>, ZHANG You dong<sup>1</sup>

(1. Coll. of Information Science and Technology, Nanjing Univ. of Aeronautics and Astronautics, Nanjing 210016, China; 2. Coll. of Electronic and Information Engineering, Nanjing Univ. of Information Science and Technology, Nanjing 210044, China)

**Abstract:** To overcome the difficulty of the speed bottleneck of collaborative filtering (CF) algorithm used for generating recommendation, a CF algorithm based on clustering basal users is presented. The algorithm separates the procedure of recommendation into offline and online phases. In the offline phase, the data of basal users are preprocessed, and the basal users are clustered; while in the online phase, the nearest neighbors of an active user are found according to the basal user clusters, and the recommendation to the active user is produced. The experimental results show that the presented algorithm can improve the performance of CF systems in both the recommendation quality and efficiency.

**Keywords:** recommendation algorithm; collaborative filtering; cluster; mean absolute error (MAE)

## 0 引 言

随着互联网的普及和电子商务的发展, 推荐系统已经逐渐成为电子商务的一项重要研究内容, 得到了越来越多研究者的关注。目前, 几乎所有大型的电子商务系统, 如 Amazon, CDNOW, eBay 等等, 都不同程度地使用了各种形式的自动推荐系统。

根据推荐的生成方式, 可以将目前的推荐系统分为两类: 基于内容的和协同推荐系统。基于内容的推荐系统是根据目标用户自己以往对项目 (如电子商务中的商品、电影、音乐等) 的喜好或评分, 判断其对目标项目的喜好程度, 从而产生推荐<sup>[1]</sup>。协同过滤推荐算法基本思想是通过计算

目标用户与各个基本用户对项目评分之间的相似性, 搜索目标用户的最近邻居, 然后由最近邻居的评分数据向目标用户产生推荐, 即目标用户对未评分项目的评分可以通过最近邻居对该项目评分的加权平均值进行逼近, 从而产生推荐<sup>[2]</sup>。

随着电子商务系统规模的扩大, 将导致用于产生推荐的数据严重稀疏, 从而影响推荐生成速度和质量。为此, 人们将统计分析、机器学习等领域的方法与协同过滤相结合, 产生了基于模型的协同过滤算法, 主要集中在以下几个方面: 通过用户对项目的评分, 对用户行为建立统计模型, 并与协同过滤系统结合<sup>[3-9]</sup>; 认为用户隶属于一个或多个潜在的类别, 将潜在语义分析引入对用户兴趣的建模分

收稿日期: 2006-04-28; 修回日期: 2006-11-18。

基金项目: 江苏省自然科学基金(BK2002091); 南京信息工程大学科研基金(Y507)资助课题

作者简介: 李涛(1978-), 男, 博士, 主要研究方向为数据挖掘、电子商务。E-mail: lthnxx@126.com

析,然后结合协同过滤进行推荐<sup>[5]</sup>;建立基于项目的协同过滤系统,引入回归模型对项目评分进行预测<sup>[6]</sup>;关联规则挖掘与协同过滤系统结合<sup>[7~8]</sup>;其他方法与协同过滤系统结合<sup>[9~11]</sup>。这些算法均部分地改进了协同过滤算法。

本文提出一种基于用户聚类的协同过滤推荐算法,通过离线对基本用户进行预处理并聚类,使得推荐系统在线产生推荐时的工作量相对变小。因而在一定程度上解决了传统的协同过滤算法在生成推荐时的速度瓶颈问题。同时实验结果表明,本文提出的算法在一定程度上能够提高推荐质量。

## 1 相关概念

协同过滤算法是根据基本用户的观点产生对目标用户的推荐列表,它基于这样的假设:如果用户对一些项目的评分比较相似,则他们对其他项目的评分也将会比较相似。协同过滤推荐系统首先搜索目标用户的若干最近邻居,然后根据最近邻居对项目的评分预测目标用户对项目的评分,从而产生推荐列表。

**定义1** 推荐系统中的数据源  $D=(U, I, R)$ , 其中  $U=\{User_1, User_2, \dots, User_m\}$  是基本用户的集合,  $|U|=m$ ;  $I=\{Item_1, Item_2, \dots, Item_n\}$  是项目集合,  $|I|=n$ ;  $m \times n$  阶矩阵  $R$  是基本用户对各项目的评分矩阵, 其中的元素  $r_{ij}$  表示  $U$  中第  $i$  个用户对  $I$  中第  $j$  个项目的评分。

在现实环境中,由于  $m$  和  $n$  的值都会比较大,用户不可能对所有的项目都逐个进行评分,因而在用户评分数据矩阵  $R(m, n)$  存在大量的空值,即对应用户没有对该项目进行评分。随着系统规模的扩大,用户数目和项目数目的迅速增加,  $R(m, n)$  中空值的数目将急剧增加,从而导致评分数据极端稀疏。为了找到目标用户的最近邻居,系统必须度量用户之间的相似性,然后选择相似性最高的若干用户作为目标用户的最近邻居。由于数据的稀疏性,以及全部用户数目庞大,导致协同过滤系统在线搜索目标用户的最近邻居的时间较长,影响用户使用效果。

在搜索最近邻居过程中,用户之间相似性的度量方法主要包括如下三种:余弦相似性、相关相似性以及修正的余弦相似性。文献[12]已经通过实验对各种度量方法进行了比较,根据其结论,我们选用 Pearson 相关相似性度量作为本文的用户相似性度量方式。

**定义2** 相关相似性(correlation based similarity),又称为 Pearson 相关相似性,设经用户  $i$  和用户  $j$  共同评分的项目集合用  $S$  表示,则用户  $i$  和  $j$  之间的相似性  $sim(i, j)$  是通过 Pearson 相关系数度量的

$$sim(i, j) = - \frac{\sum_{u \in S} (r_{iu} - \bar{r}_i)(r_{ju} - \bar{r}_j)}{\sqrt{\sum_{u \in S} (r_{iu} - \bar{r}_i)^2} \sqrt{\sum_{u \in S} (r_{ju} - \bar{r}_j)^2}} \quad (1)$$

其中,  $r_{iu}$  表示用户  $i$  对项目  $u$  的评分,  $r_{ju}$  表示用户  $j$  对项目  $u$  的评分,  $\bar{r}_i$  和  $\bar{r}_j$  分别表示用户  $i$  和用户  $j$  对项目的平均评分。

**定义3** 已知数据源  $D=(U, I, R)$ , 给定目标用户  $a$ , 及其对  $I$  中项目评分向量  $A(1, n)$ , 对于  $\forall i \in U$ , 将  $sim(a, i)$  最大的  $l$  个基本用户  $i$  组成集合  $NS_a$ , 则称该集合中的元素为目标用户  $a$  的最近邻居。

**定义4** 已知数据源  $D=(U, I, R)$ , 给定目标用户  $a$ , 及其最近邻居集合  $NS_a$ , 则用户  $a$  对项目  $j$  的预测评分  $P_{aj}$  为

$$P_{aj} = \bar{r}_a + \frac{\sum_{i \in NS_a} sim(a, i) \times (r_{ij} - \bar{r}_i)}{\sum_{i \in NS_a} |sim(a, i)|} \quad (2)$$

其中,  $\bar{r}_a$  和  $\bar{r}_i$  分别表示用户  $a$  和用户  $i$  对项目的平均评分。

## 2 基于用户聚类的协同过滤算法

本文提出基于用户聚类的协同过滤算法,分为离线和在线两部分。离线时,算法首先对基本用户聚类,产生若干用户聚类中心,然后计算每一个用户与各聚类中心相似性度量,从而获得基本用户与各个聚类中心的相似性度量矩阵。在线时,算法仅仅计算目标用户与各个聚类中心的相似性,然后通过对比离线时获得的相似性度量矩阵,搜索目标用户的最近邻居,再由这些最近邻居进行目标用户评分的预测,并产生推荐。由于离线算法获得的用户聚类数目远远少于用户数目,在线时系统只需要计算目标用户与少量的聚类中心的相似性,因而,在线查找目标用户的最近邻居所需时间将大大缩短。

### 2.1 离线用户聚类算法

本文算法对基本用户进行聚类的目的是产生基本用户的类别所属程度矩阵,即基本用户与各聚类中心之间的相似性矩阵,使得系统能够在线时通过类别所属程度矩阵快速搜索到目标用户的最近邻居,因而本文算法对聚类的精确程度要求并不是很高,只需要获得大多数元素的聚类即可。对于少量没有进入聚类类别的用户,我们只需获得其对各聚类中心的相似性。

目前的聚类方法有很多,本文采用 k Means 算法<sup>[13]</sup>的思想对基本用户进行聚类。由于在使用 k Means 算法时需要事先给出聚类数目,因而本文首先采用对基本用户进行预处理以确定聚类数目。在此过程中,首先计算基本用户两两之间的相似性,将相似性大于一定阈值的基本用户归于同一原始类别中。然后选取包含用户数量最多的前  $k$  个原始类别作为初始条件,通过它们计算出  $k$  个初始聚类中心,进而进一步聚类。

**算法1** 基本用户聚类算法。

输入:用户聚类的数目  $k$  和数据源  $D=(U, I, R)$ 。

输出: $k$  个用户聚类中心  $Cluster(k, n)$ , 类别所属程度矩阵  $V(m, k)$ 。

方法:

- 1) 经过预处理计算产生的  $k$  个原始类别;
- 2) 每个原始类别中任选一个对象作为初始的聚类中心;
- 3) repeat
- 4) for  $i=1; m\{$
- 5) 计算用户  $i$  与各个聚类中心之间的相似性  $v_{i1}, v_{i2}, \dots, v_{ik}$ ;
- 6) 取出这些相似性中最大值;
- 7) if (该最大值  $>$  相似性阈值) {
- 8) 将用户  $i$  所属类别定为该聚类;
- 9) }
- 10) }
- 11) 同一聚类的所有用户平均评分作为该聚类的聚类中心, 重新计算每个聚类中心;
- 12) until 聚类中心不再发生变化。

由算法 1 获得用户聚类中心  $Cluster(k, n)$  以及类别所属程度矩阵  $V(m, k)$ 。

用户聚类中心  $Cluster(k, n)$  中,  $k$  行代表  $k$  个用户聚类中心,  $n$  列代表  $n$  个项目, 第  $i$  行第  $j$  列的元素  $rc_{ij}$  代表用户聚类中心  $i$  对项目  $j$  的评分, 其是用户聚类  $i$  中所有用户对项目  $j$  评分的均值。

基本用户的类别所属程度矩阵  $V(m, k)$  中,  $m$  行代表  $m$  个基本用户,  $k$  列代表  $k$  个用户聚类, 第  $i$  行第  $j$  列的元素  $v_{ij}$  代表用户  $i$  对于聚类中心  $j$  的相似性, 即用户  $i$  和第  $j$  个用户聚类中心之间的 Pearson 相关相似性度量。经过聚类, 聚类中心数目能够远远小于基本用户的数目, 即  $k \ll m$ 。同时, 在实际系统中, 聚类中心数目能够远远小于项目的数目, 即  $k \ll n$  一般总会成立。

## 2.2 在线搜索最近邻居并产生推荐

本文提出的算法在上述离线处理结果的基础上, 首先计算目标用户与各个聚类中心之间的相似性, 获得目标用户所属各个聚类程度的向量, 然后搜索类别所属程度矩阵, 确定目标用户的最近邻居。在线搜索近邻算法如算法 2 所示。

**算法 2** 在线搜索目标用户最近邻居算法。

输入: 用户聚类中心矩阵  $Cluster(k, n)$ , 目标用户评分向量, 类别所属程度矩阵  $V(m, k)$ 。

输出: 目标用户的  $l$  个最近邻居。

方法:

- 1) 计算目标用户与  $k$  个聚类中心之间的相似性, 获得  $1 \times k$  的向量  $(v_1, v_2, \dots, v_k)$ ;
- 2) 计算向量  $(v_1, v_2, \dots, v_k)$  与类别所属程度矩阵  $V(m, k)$  各行之间的欧氏距离;
- 3) 上述欧氏距离最小的前  $l$  个基本用户视为目标用户的最近邻居;

通过算法 2, 我们得到目标用户的最近邻居后, 下一步

可以直接根据公式 (2) 产生对目标用户的推荐。

## 3 算法的时间复杂度分析

在实际的推荐系统中, 在线产生推荐的时间复杂度是衡量推荐算法的主要指标, 因而, 此处主要分析基于用户聚类的协同过滤算法的在线时间复杂度, 并传统的协同过滤的在线算法时间复杂度进行比较。

协同过滤算法在线产生推荐的过程主要分为如下几步: 首先计算目标用户与  $m$  个基本用户之间的相似度, 然后对这组相似性排序以查找到最近邻居, 最后根据查找到的最近邻居产生推荐。

传统的协同过滤算法直接计算目标用户与所有的  $m$  个基本用户之间的相似性。为了计算目标用户与当前基本用户之间的相似性, 首先需要查找共同评分项目, 为此需要对比  $n$  个项目以确定共同评分项目。因而, 计算目标用户与所有的  $m$  个基本用户之间的相似性的时间复杂度为  $O(m \times n)$ , 由于  $m, n$  为同一数量级, 故时间复杂度为  $O(n^2)$ 。

基于用户聚类的协同过滤推荐算法首先计算目标用户与基本用户聚类中心之间的相似性得到向量  $(v_1, v_2, \dots, v_k)$ , 然后计算该向量与类别所属矩阵各行之间的欧氏距离。其中, 计算与聚类中心相似性的时间复杂度为  $O(k \times n)$ , 计算各欧氏距离的时间复杂度为  $O(k \times m)$ , 所以整个时间复杂度为  $O(k \times n) + O(k \times m)$ 。由于  $m, n$  为同一数量级, 而聚类数量  $k$  为远小于  $m, n$  的常数, 所以时间复杂度为  $O(n)$ 。

计算出目标用户与  $m$  个基本用户的相似度之后, 两类算法均对  $m$  个相似度进行排序, 查找出最近邻居, 时间复杂度为  $O(n \log n)$ ; 然后通过  $l$  个最近邻居产生推荐, 时间复杂度为  $O(l \times n)$ , 由于  $l$  为远小于  $n$  的常数, 故时间复杂度为  $O(n)$ 。所以, 总体上来看, 两类算法的时间复杂度主要区别在计算相似性上, 而且基于用户聚类的算法要优于传统的算法。

## 4 实验结果及分析

我们通过实验测试了本文所提出的算法性能, 并与其他一些协同过滤算法进行了比较。

### 4.1 实验环境及实验数据

性能测评的硬件平台是配置 Intel Pentium IV 2.4GHz CPU 和 1G RAM 的 PC, 操作系统运行 Windows XP Professional, 所有程序均采用 Matlab 实现。

实验数据采用美国 Minnesota 大学 GroupLens 项目组提供的 MovieLens 数据集 (<http://www.grouplens.org/>)。MovieLens 是 GroupLens 项目组开发的一个基于 Web 的研究型推荐系统, 用于接收用户对电影的评分并提供相应的电影推荐列表。目前, 该 Web 站点的用户已经超过 43 000 人, 可供用户评分的电影超过 3 500 部。MovieLens

数据集中包含了 943 个用户对 1 682 部电影的 100 000 条评分数据,其中每个用户至少对 20 部电影进行了评分。GroupLens 项目组提供全部 MovieLens 数据集的同时,将其分成 5 个互不相交的子集,然后每次选择一个子集作为 test 数据集,其他四个合为一个 base 数据集,从而形成了 5 对 base 数据集和 test 数据集。在此基础上,我们使用 5-折交叉验证(5 fold cross validation)方法进行实验。每次选择一对 base 数据集和 test 数据集,使用 base 数据集中的记录作为基本用户,对 test 数据集中的目标用户进行推荐测试。5-折交叉验证之后计算误差的均值作为实验算法的误差结果。

#### 4.2 度量标准

评价推荐系统质量的度量标准主要包括统计精度度量方法和决策支持精度度量方法<sup>[6]</sup>。统计精度度量方法中的平均绝对误差(mean absolute error, MAE)易于计算和理解,可以直观地对推荐质量进行度量,是最常用的一种推荐质量度量方法,本文也采用 MAE 作为度量标准。

定义 5 设目标用户的预测评分集合为  $\{p_1, p_2, \dots, p_N\}$ , 对应的实际评分集合为  $\{q_1, q_2, \dots, q_N\}$ , 则平均绝对误差定义为

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (3)$$

可见,MAE 是通过计算目标用户的预测评分与实际评分之间的偏差度量预测的准确性,因而 MAE 越小,推荐质量越高。

#### 4.3 实验结果

为了验证本文提出的基于用户聚类的协同过滤算法(collaborative filtering based on clustering basal users, UCCF)的有效性,我们进行了实验对比。参与对比实验的算法包括:传统的协同过滤算法(traditional collaborative filtering, CF)<sup>[2]</sup>, Marlin 提出的用户评分特性模型协同过滤算法(user rating profile, URP)<sup>[3]</sup>, Hofmann 提出的潜在语义模型协同过滤算法(latent semantic models, LSM)<sup>[5]</sup>, Sarwar 等提出的基于项目的协同过滤算法(item based collaborative filtering, IBCF)<sup>[6]</sup>等。

在本文算法验证过程中,首先需要确定预处理以及算法 1 中的相似性阈值,在整个实验过程中,我们选取该阈值为 0.5。其次需要确定聚类数量  $k$ ,通过对 MovieLens 提供的数据进行预处理发现,原始类别的前 7 类中,每个原始类别都包含相对较多的基本用户,而其它类别则只包含少量的基本用户,所以我们首先选取  $k=7$  进行实验。由于在本文所提出的算法中,聚类的目的是便于用户在线查找最近邻居,而对聚类本身的要求并不苛刻,因而还分别选取了  $k=5, 6, 8$  进行实验。

中的一部分项目进行评分。因而,在计算用户之间的相似性时,需要考虑用户之间共同评分的项目的数量。在整个实验过程中,如果二者之间共同评分项目的数量低于某个阈值时,我们认为二者无可比性,直接将二者之间的相似性置为 0。在实验过程中,我们首先将共同评分项目阈值设定为 20,即用户之间共同评分项目的数量大于 20 时,我们计算其 Pearson 相关相似性,否则直接将其相似性置为 0。目标用户的最近邻居数量  $l$  从 10 变化到 30,间隔为 5,然后分别对参与对比实验的算法和  $k=5, 6, 7, 8$  的 UCCF 算法进行对比实验,结果如图 1 所示。

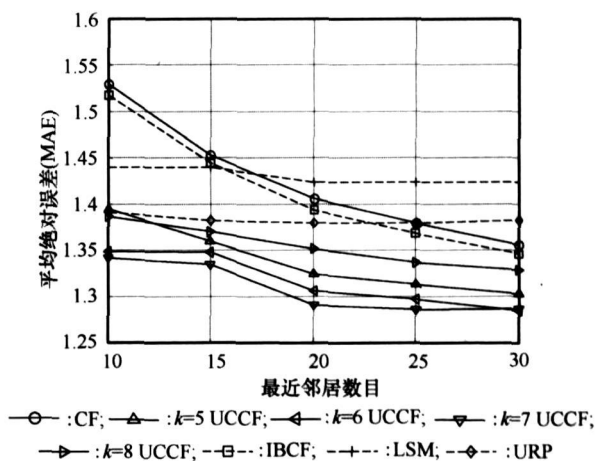


图 1 共同评分项目阈值固定为 20 时推荐精度比较

进一步实验过程中,我们将目标用户的最近邻居数量  $l$  设定为 20。计算用户之间相似性时,共同评分项目数量阈值从 10 变化到 30,间隔为 5,再次进行对比实验,结果如图 2 所示。

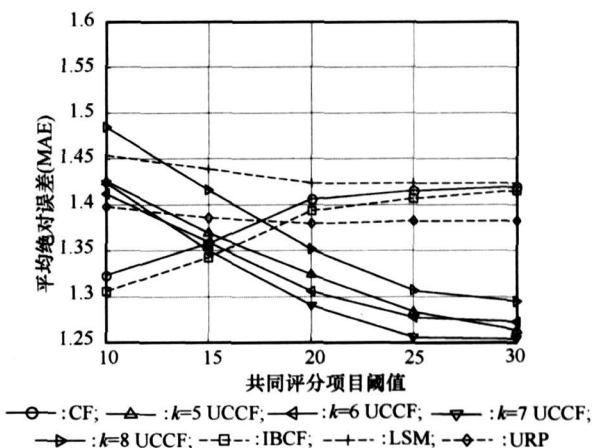


图 2 最近邻居阈值固定为 20 时推荐精度比较

由图 1 和图 2 可以看出,在相同条件下,不论是固定共同评分项目阈值,变动最近邻居数量,还是固定最近邻居数量,变动共同评分项目阈值,一般而言, UCCF 算法的 MAE 均在一定程度小于其他对比算法。可见,本文提出的算法

可以较为显著地提高推荐系统的推荐质量。

#### 4.4 实验结果分析

由前文给出的算法时间复杂度分析可知,本文提出的基于用户聚类的协同过滤推荐算法能够提高在线搜索目标用户的最近邻居以及推荐生成的速度,保证了推荐系统的实时性要求。实验过程中,我们确实发现推荐生成时间大大缩短,此处不再给出实验结果。由于 IBCF 算法是使用项目之间的相似度参与推荐,在评分矩阵相同的情况下,与传统的协同过滤算法直接使用用户之间的相似度参与推荐在本质上没有改变,因而实验结果也看出其误差趋势基本一致。对于 LSM 和 URP 算法,它们都是首先建立了用于描述用户特性的模型,然后通过模型参与和产生推荐,因而推荐质量体现出了对共同评分项目数量及近邻数量不太敏感的特性,从实验结果也看出这一结论。从整体结果来看,本文提出的算法的推荐质量在一定程度上是优于这些对比算法的。

同时,相对与 LSM 及 URP 等算法而言,UCCF 算法无需建立描述用户特性的模型,更加易于理解和实现。

此外,我们注意到,UCCF 算法对基本用户进行聚类的目的是为了在线算法查找最近邻居,对用户聚类的精确度并没有苛刻的要求。实验也表明,即使用户聚类并没有达到最优,算法仍然能够在一定程度上加快推荐生产速度,提高推荐质量。

## 5 结 论

针对传统协同过滤推荐算法推荐生成速度问题,本文提出了基于用户聚类的协同过滤推荐算法,通过必要的离线数据运算,提高了在线时推荐的生成速度。实验结果表明,不仅能够一定程度上解决生成推荐的速度问题,而且能够提高推荐质量。当然,基于用户聚类的协同过滤算法还有一些问题需要进一步研究,例如聚类过程中  $k$  值的选取原则等。

## 参考文献:

[1] Balabanovic M, Shoham Y. Fab: content based collaborative recommendation[J]. *Communications of the Association for Computing Machinery*, 1997, 40(3): 66-72.

[2] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of Netnews[C] // *Proc. of ACM 1994 Conference on Computer Supported Cooperative Work*. Chapel Hill: ACM Press, 1994: 175-186.

[3] Marlin B. Modeling user rating profiles for collaborative filtering[C] // *Advances in Neural Information Processing Systems 16 (NIPS16)*. Cambridge, MA: MIT Press, 2003: 627-634.

[4] Yu K, Schwaighofer A, Tresp V, et al. Probabilistic memory based collaborative filtering[J]. *IEEE Transaction on Knowledge and Data Engineering*, 2004, 16(1): 56-69.

[5] Hofmann T. Latent semantic models for collaborative filtering[J]. *ACM Trans. Information Systems*, 2004, 22: 89-115.

[6] Sarwar B, Karypis G, Konstan J, et al. Item based collaborative filtering recommendation algorithm[C] // *Proc. of the 10th International World Wide Web Conference*. New York: ACM Press, 2001: 285-295.

[7] Leung C W, Chan S C, Chung F. A collaborative filtering framework based on fuzzy association rules and multiple level similarity[J]. *Knowledge Information Systems*. 2006, 9(4): 492-511.

[8] Lin W Y, Alvarez S A, Ruiz C. Efficient adaptive support association rule mining for recommender systems[J]. *Data Mining and Knowledge Discovery*. 2002, 6(2): 83-105.

[9] Si L, Jin R. Flexible mixture model for collaborative filtering[C] // *Proc. of the 20th International Conference on Machine Learning*. Washington, DC: AAAI Press, 2003: 704-711.

[10] Basilico J, Hofmann T. Unifying collaborative and content based filtering[C] // *Proc. of the 21st International Conference on Machine Learning*. New York: ACM Press, 2004: 65-72.

[11] Jason R, Nathan S. Fast maximum margin matrix factorization for collaborative prediction[C] // *Proc. of the 22nd International Conference on Machine Learning*. New York: ACM Press, 2005: 713-719.

[12] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C] // *Fourteen Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin: Morgan Kaufmann, 1998: 43-52.

[13] Han J W, Kamber M. 数据挖掘: 概念与技术[M]. 北京: 机械工业出版社, 2001: 232-235.