基于用户兴趣度和特征的优化协同过滤推荐

严冬梅, 鲁城华
(天津财经大学 理工学院 信息科学与技术系, 天津 300222)

摘 要: 协同过滤技术目前被广泛应用于个性化推荐系统中。为了使用户的最近邻居集合更加精确有效,提出了基于用户兴趣度和用户特征的优化协同过滤推荐算法。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分析出用户具有不同特征时对项目的喜好程度;最后采用一种新的相似度度量方法计算出目标用户的最近邻居集合。实验表明该算法提高了最近邻居集合的有效性和准确度,推荐质量较以往算法有明显提高。

关键词: 用户兴趣度; 用户特征; 贝叶斯算法; 协同过滤; 用户相似度
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2012)02-0497-04
doi:10.3969/j.issn.1001-3695.2012.02.026

Optimized collaborative filtering recommendation based on users' interest degree and feature

YAN Dong-mei, LU Cheng-hua
(*Dept. of Information Science & Technology, Institute of Technology, Tianjin University of Finance & Economics, Tianjin* 300222, *China*)

Abstract: Collaborative filtering technology is widely used in personalized recommendation system. In order to make the user's nearest neighbors set more precise and effective, this paper presented an optimized collaborative filtering recommendation algorithm based on users' interest degree and feature. Firstly, it grouped users through calculating users' interest degree to items. Secondly, it got the value of the users' preferences for items when the users had different characteristics. Finally, it used a new method of calculating the similarity degree to calculate the target users' nearest neighbors set. The result shows that the algorithm enhances the effectiveness and accuracy of the nearest neighbors set, and the recommendation quality has significant improvement than traditional algorithm.

Key words: users' interest degree; users' feature; Bayesian algorithm; collaborative filtering(CF); similarity between users

0 引言

随着互联网的发展,电子商务技术日渐成熟并广泛应用到大众生活当中。人们越来越喜欢在网络上寻找自己所需要的信息,因此各个网站研究的热点开始转向于根据网站自身已有的网络用户信息来推荐他们感兴趣的信息[1]。协同过滤(CF)是最成功的个性化推荐技术,它借助已知的用户评价来实现对目标用户的推荐。典型的协同过滤推荐算法是基于用户的协同过滤推荐算法,其基本原理是利用历史评分数据形成用户邻居,根据评分相似的最近邻居的评分数据向目标用户产生推荐[2]。随着用户和项目数目的增多,如何提高算法的可扩展性和推荐质量是协同过滤技术面临的主要问题。对此,很多学者提出了解决方案,文献[3]提出了基于模式协同过滤算法,利用朴素的贝叶斯网络、NB-ELR、tree-argument NB 或者 TAN-ELR 分类器增强处理稀疏数据方面的能力,对用户—项目矩阵中缺失的评分实现填充。该方法对解决数据稀疏性有一定的效果,但对算法性能的改变意义不大。文献[4]提出通过奇异值分解减少项目空间的维数,使用户在降维后的项目空间上对每个项目均有评分。此方法能有效提高推荐系统的伸缩能力,但是降维会导致信息丢失,尤其是项目的空间维数很高时,降维效果难以保证。文献[5]把项目/评分对映射成 DHT 的键值,并且允许查找评分相似的用户,此方法有好的可扩展性,但是要在有评分集合的前提下才能产生推荐。

从传统推荐算法中可看出,构造有效的用户邻居集合是提高推荐质量的关键。本文提出基于用户兴趣度和特征的优化协同过滤推荐算法,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分析出用户具有不同特征时其对项目的喜好程度;最后采用优化的相似度度量方法计算出用户的最近邻集合,形成推荐。实验表明该算法的推荐质量较以往算法有明显提高。

1 用户相似度的计算

基于用户的协同过滤推荐是基于这样一个假设: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比较相似[6]。推荐过程可分为三步:a)首先获得用户对项目的评分,建立用户—项目评分矩阵;b)计算目标用户与其他各个用

收稿日期: 2011-08-02; **修回日期**: 2011-09-29

作者简介: 严冬梅(1972-),女,浙江萧山人,副教授,博士,主要研究方向为软件工程、计算机系统分析与设计、管理信息系统与决策支持系统(ydongmei326@gmail.com);鲁城华(1985-),女,硕士研究生,主要研究方向为信息与计算科学、计算机应用技术、管理信息系统与决策支持系统.

户的相似度,得到目标用户的最近邻居集合;c)根据其最近邻居对项目的评分信息预测目标用户对未评分项目的评分,取出预测评分最高的前 n 项推荐给用户,完成推荐[7]。

1.1 建立用户—项目评分矩阵

用户—项目评分矩阵是一个 m 行 n 列的矩阵,第 i 行第 j 列的元素 R_{ij} 代表用户 i 对项目 j 的评分[8]。用户—项目评分矩阵 $R(m,n)$ 如表 1 所示。

表 1 用户—项目评分矩阵 $R(m,n)$

项目用户	$item_1$	…	$item_j$	…	$item_n$
$user_1$	$R_{1,1}$	…	$R_{1,j}$	…	$R_{1,n}$
…	…	…	…	…	…
$user_i$	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,n}$
…	…	…	…	…	…
$user_m$	$R_{m,1}$	…	$R_{m,j}$	…	$R_{m,n}$

表中 m 为用户数量,n 为项目数量,矩阵元素 $R_{i,j}$ 的评估值与项的内容有关,如果项是商品,则表示用户订购与否,例如 1 表示订购,0 表示没有订购;如果项是 Web 文档,则表示浏览与否,用户对它的兴趣有多高。评估值可以按照等级来分,如 15,评估值越大,说明用户对项目的喜爱程度越高。

1.2 计算用户相似度

度量用户 u 和 v 的相似性的方法是,首先获得用户 u 和 v 评分的所有项目,然后通过相似性度量方法计算用户 u 与 v 之间的相似性,记为 $\mathrm{sim}(u,v)$。计算相似性的算法有很多,常用的方法有三种:

a)Pearson 相关相似性。用户 u 和 v 共同评分过的项目集合用 P_{uv} 表示,$R_{u,a}$、$R_{v,a}$ 分别表示用户 u 和 v 对项目 a 的评分,$\overline{R_u}$ 和 $\overline{R_v}$ 分别表示用户 u 和 v 的平均评分,公式为

$$\mathrm{sim}(u,v)=\frac{\sum_{a\in P_{uv}}(R_{u,a}-\overline{R_u})(R_{v,a}-\overline{R_v})}{\sqrt{\sum_{a\in P_{uv}}(R_{u,a}-\overline{R_u})^2}\sqrt{\sum_{a\in P_{uv}}(R_{v,a}-\overline{R_v})^2}}$$

b)余弦相似性。设用户 u 和 v 在 n 维项目空间上的评分分别表示为向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$,则它们之间的相似性 $\mathrm{sim}(\boldsymbol{u},\boldsymbol{v})$ 为

$$\mathrm{sim}(u,v)=\cos(\boldsymbol{u},\boldsymbol{v})=\frac{\boldsymbol{u}\cdot\boldsymbol{v}}{\|\boldsymbol{u}\|\times\|\boldsymbol{v}\|}$$

分子为两个用户评分向量的内积,分母为两个用户向量模的乘积。

c)修正的余弦相似性[2]。修正方法中考虑了不同用户的评分尺度问题,用户 u 和 v 共同评分过的项目集合用 P_{uv} 表示,P_u 和 P_v 分别表示用户 u 和 v 评分过的项目集合;$R_{u,a}$、$R_{v,a}$ 分别表示用户 u 和 v 对项目 a 的评分;$\overline{R_u}$ 和 $\overline{R_v}$ 分别表示用户 u 和 v 的平均评分,公式为

$$\mathrm{sim}(u,v)=\frac{\sum_{a\in P_{uv}}(R_{u,a}-\overline{R_u})(R_{v,a}-\overline{R_v})}{\sqrt{\sum_{a\in P_u}(R_{u,a}-\overline{R_u})^2}\sqrt{\sum_{a\in P_v}(R_{v,a}-\overline{R_v})^2}}$$

本文中采用第三种方法计算用户 u 与其他用户之间的相似度,但是这个公式中只考虑到被用户 u 和 v 共同评过分的项目集合,并没有考虑集合当中的项目个数。为了使计算结果更加精确,本文引入一个阈值 θ 进行调整,用公式表示为

$$\mathrm{sim}_L(u,v)=\frac{Q}{\mathrm{MAX}(\theta,Q)}\cdot\mathrm{sim}(u,v)$$

其中:Q 表示被用户 u 和 v 共同评过分的项目个数;θ 为设定的阈值;$\mathrm{MAX}(\theta,Q)$ 表示取 θ 和 Q 中数值较大的一个,如果被用户 u 和 v 共同评过分的项目个数大于或等于 θ,则不需要调整评分相似性,仍采用原来的相似度结果;反之,则取 θ,用 Q/θ 来调整用户的评分相似性。本文将在下面章节中对用户相似度的计算进行进一步的优化调整。

2 基于贝叶斯算法的协同过滤推荐技术

2.1 用户对项目的兴趣度

本文中引入了用户 v 对项目 i 的兴趣度大小的计算,设为 $p(v,i)$。设对项目 i 进行评分的用户集合为 U_i,对于用户 $v\in U_i$,如果 v 和 U_i 中用户相似度比较高,则说明用户 v 对项目 i 的感兴趣程度比较大,则兴趣度值 $p(v,i)$ 就比较高。因此定义 $p(v,i)$ 来衡量用户 v 对项目 i 的感兴趣程度,定义为

$$p(v,i)=\frac{\sum_{a\in U_i}\mathrm{sim}(v,a)}{|U_i|}$$

其中:$\mathrm{sim}(v,a)$ 是用户 v 与 a 的相似度值;$|U_i|$ 为用户数目。若 U_i 中用户数为 n,则计算 $p(v,i)$ 的时间复杂度为 $O(n^2)$,由于访问同一个项目的用户数一般不大,因此对算法运行效率影响不大。

2.2 用户特征属性

实际生活中,不同特征的人拥有的兴趣爱好可能不同,而同一类别的人的兴趣爱好则具有一定的相似性,因此往往具有相似特征的用户兴趣取向也具有相似性。通过分析用户的不同特征对项目的选择偏好,有利于更加精确地向目标用户产生推荐。

用户的特征包括年龄、性别、职业、经济条件、生活方式、个性、自我概念以及在生命周期中所处的阶段等各方面特征,在本文中将这些特征抽象为 $\{R_1,R_2,\cdots,R_n\}$,并假设各属性之间是相互独立的。将所有用户特征进行编码,编码唯一,不同用户的不同特征属性对应不同的编码。用矩阵表示如表 2 所示。

表 2 用户特征值矩阵

用户	R_1	R_2	…	R_n
user1	100000	200001	…	300001
user2	100001	200010	…	300010
…	…	…	…	…
user3	100010	200011	…	300011

2.3 利用贝叶斯算法分析用户特征值

根据前边计算的用户对项目的兴趣度来对项目 i 进行评分的用户集合进行分组,分为喜欢组和不喜欢组,即对项目 i 兴趣度大的用户放在喜欢组;反之放在不喜欢组,分别设为 P^{like} 和 P^{unlike}。然后利用贝叶斯算法分析用户不同特征值在喜欢组和不喜欢组当中出现的概率,以此来计算当用户具有某些特征值时对未评分项目喜欢的概率。前边已经计算出用户对项目 i 的兴趣度 $p(v,i)$,如果 $p(v,i)>0.5$,则 $u\in P^{\mathrm{like}}$,反之 $u\in P^{\mathrm{unlike}}$。

用 $r_1,r_2,\cdots,r_n$ 代表用户特征值,假设事件 M 是喜欢项目 i 的用户集合,事件 N 是不喜欢项目 i 的用户集合,$P_i(r_k|M)$ 表

示喜欢项目 i 的用户集合中出现特征值 r_k 的概率,$P_i(r_k|N)$ 表示不喜欢项目 i 的用户集合中出现特征值 r_k 的概率。然后对每个项目建立一个哈希表,Hash_like 对应喜欢项目 i 的用户集合,Hash_unlike 对应不喜欢项目 i 的用户集合,两个哈希表分别存放用户特征值及其在对应的集合中出现概率的映射关系。$P_iu(M|r_k)$ 表示用户 u 具有特征值 r_k 时喜欢项目 i 的概率。$P_iu(M|r_1,r_2,\cdots,r_n)$ 表示用户同时具有特征值 $r_1,r_2,\cdots,r_n$ 时喜欢项目 i 的概率。根据贝叶斯公式可得出

$$P_i^u(M|r_k)=\frac{P_i(r_k|M)\cdot P_i(M)}{P_i(r_k|M)\cdot P_i(M)+P_i(r_k|N)\cdot P_i(N)}$$

当 $P_i(M)$ 和 $P_i(N)$ 都取值为 0.5 时,此公式可以简化为

$$P_i^u(M|r_k)=\frac{P_i(r_k|M)}{P_i(r_k|M)+P_i(r_k|N)}$$

假设目标用户 u 具有的特征值为 $r_1,r_2,\cdots,r_n$,并设一个集合 $W=\{w_1,w_2,\cdots,w_n\}$。若 r_k 已经在计算出来的哈希表中,就把 r_k 加入集合 W 中。如果集合 W 为空,那么 $P_iu(M|w_1,w_2,\cdots,w_n)$ 的取值为 0.5;如果集合 W 不为空,那么由复合概率公式可以得到:

$$P_i^u(M|w_1,w_2,\cdots,w_n)=$$
$$\{P_i^u(M|w_1)P_i^u(M|w_2)\cdots P_i^u(M|w_n)\}/\{P_i^u(M|w_1)P_i^u(M|w_2)\cdots$$
$$P_i^u(M|w_n)+[1-P_i^u(M|w_1)][1-P_i^u(M|w_2)]\cdots[1-P_i^u(M|w_n)]\}$$

此结果为具有特征值 $w_1,w_2,\cdots,w_n$ 的用户 u 喜欢未评分项目 i 的概率值。

2.4　求出用户的最近邻居集合

在前文中已经计算出了用户相似度 $\mathrm{sim}_L(u,v)$,计算结果相对传统相似度计算在精确度上提升了一些。为了使精确度更高,将 $P_i^u(M|w_1,w_2,\cdots,w_n)$ 概率值加入相似度的计算中,定义最终相似度公式如下:

$$\mathrm{SIM}(u,v)=\delta\,\mathrm{sim}_L(u,v)$$

其中:δ 为调节因子。其来源如下:

$$\delta=\begin{cases}P_i^u & P_i^u>0.5\ \text{且}\ p(v,i)>0.5\\ 1-P_i^u & P_i^u>0.5\ \text{且}\ p(v,i)<0.5\\ P_i^u & P_i^u<0.5\ \text{且}\ p(v,i)>0.5\\ 1-P_i^u & P_i^u<0.5\ \text{且}\ p(v,i)<0.5\end{cases}$$

其中:$P_i{}^u$ 表示用户 u 喜欢项目 i 的概率,$p(v,i)$ 表示用户 v 对项目 i 的兴趣度。这样可以使同时喜欢项目 i 的两个用户或者同时不喜欢项目 i 的两个用户具有更高的相似度,取相似度最高的若干用户组成目标用户的邻居集合,从而构造出更加有效的邻居集合。

3　算法的详细步骤

输入:目标用户 u,用户—项目评分矩阵 $R(m,n)$,用户特征值矩阵 L。

输出:目标用户 u 的 Top-N 推荐集。

采用以下步骤计算用户 u 对未评分项目 i 的预测评分:

a)首先根据用户—项目评分矩阵 $R(m,n)$ 计算其他用户 v 与目标用户 u 的相似度 $\mathrm{sim}(u,v)$。

b)应用阈值 θ 对相似度 $\mathrm{sim}(u,v)$ 进行调整,计算出更加精确的相似度 $\mathrm{sim}_L(u,v)$。

c)计算用户 v 对已评分项目 i 的兴趣度 $p(v,i)$。

d)利用 c)中的兴趣度 $p(v,i)$ 对用户偏好进行学习,按照对项目 i 的不同兴趣度建立哈希表,hash_like 表对应喜欢项目 i 的用户集合,hash_unlike 表对应不喜欢项目 i 的用户集合,分别存放用户特征值及其在对应的集合中出现概率的映射关系。

e)用贝叶斯算法计算 P_i^u,即用户 u 喜欢项目 i 的概率。

f)用最终的相似度计算方法计算用户 u 与其他用户 v 的相似性 $\mathrm{SIM}(u,v)$。

g)将相似性最高的前 K 个项作为用户 u 的最近邻居集合,记为 K。

h)采用下面公式计算用户 u 对未评分项目 i 的预测评分 $P_{u,i}$。

$$P_{u,i}=\overline{R_u}+\frac{\sum_{v\in K}\mathrm{SIM}(u,v)\cdot(R_{v,i}-\overline{R_v})}{\sum_{v\in K}\mathrm{SIM}(u,v)}$$

其中:$\overline{R_u}$ 和 $\overline{R_v}$ 分别表示用户 u 和 v 的平均评分;K 是用户 u 的最近邻居集合;$\mathrm{SIM}(u,v)$ 是用户 u 与最近邻居集合中用户 v 的相似度值;$R_{v,i}$ 是用户 v 对项目 i 的评分。

i)经过计算,把预测评分值最高的前 n 个项目推荐给用户 u,完成推荐,算法结束。

下面对本文算法的时间复杂度进行分析,推荐系统中有 m 个用户,每个用户对 n 个项目进行评分。在算法步骤 a)中计算两两用户之间的相似度,其计算复杂度为 $O(m^2\cdot n)$,由于此计算过程可以离线进行,因此省略了在线计算的时间,提高了推荐效率;步骤 c)中计算用户 v 对已评分项目 i 的兴趣度 $p(v,i)$,由上面的计算公式可以发现,假设 U_i 中用户数为 L,则计算 $p(v,i)$ 的时间复杂度为 $O(L^2)$;步骤 d)中对用户偏好进行学习,建立哈希表,计算复杂度由用户特征值的个数决定,假设用户特征值个数为 C,则建立两个哈希表的计算复杂度都是 $O(C)$;同样,在步骤 e)中计算用户 u 喜欢项目 i 的概率时其复杂度也为 $O(C)$。总的来说,算法的计算复杂度为 $O(L^2)+O(C)$,由于访问同一个项目的用户数 L 一般不大,因此 $O(L^2)$ 对算法运行效率影响不大,算法的整体运行效率还是比较高的。

4　算法实验及分析

4.1　实验数据及评估准则

实验数据采用美国 Minnesota 大学 GroupLens 项目组提供的 Movielens 数据集 ml-data。Movielens 是一个基于 Web 的研究型推荐系统,ml-data 数据集中包含了 943 名用户对 1 682 部电影的 100 000 条打分数据,评价分值是 15 的整数,数值越大,表明用户对该电影的偏爱程度越高,每个用户至少对 20 部电影进行了评价。本文从数据集 ml-data 中随机抽取了 300 位用户对 600 部电影的评分数据,根据网站的人口统计信息整理出这 300 名用户的特征描述矩阵。

实验采用平均绝对误差(mean absolute error, MAE)作为度量算法好坏的标准。MAE 是协同过滤算法中统计精度的常用评价标准,它通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性,MAE 越小,则推荐质量

越高[9]。预测的评分数据集合表示为 $P_u=(p_{u,i}\mid i=1,2,\cdots,n)$,对应的目标用户的实际评分集合表示为 $R_u=(r_{u,i}\mid 1,2,\cdots,n)$。

$$\mathrm{MAE}=\frac{\sum_{i\in Q_u}|p_{u,i}-r_{u,i}|}{|Q_u|}$$

其中:Q_u 为测试集内目标用户 u 的预测值和真实评价值均不为 0 的项目集合。

4.2　实验设计与结果分析

本文从数据集中随机抽取了 300 位用户对 600 部电影的评分数据,实验将数据集中的数据分为训练集和测试集,设计了三组实验。

实验 1　阈值 θ 的引入使用户相似度的计算结果更加精确,本实验中将阈值 θ 设为不同的值,设定用户最近邻居数目为 30,同时将实验数据分为 4 组,Data1、Data2、Data3 为训练集,Data4 为测试集,计算其 MAE,实验结果如图 1 所示。

从实验结果可看出,如果 θ 取值太大,甚至可能会消弱用户的评分相似性,当 θ=70 时,在不同的实验数据集中 MAE 基本达到了最低。

实验 2　在此实验中测试在不同的训练集和测试集比例的情况下,以传统基于用户的 CF 算法为对照检验本文提出的算法的有效性,即验证算法在不同数据稀疏度上的性能,实验中设定用户最近邻居数目为 30,θ =70,实验结果如图 2 所示。

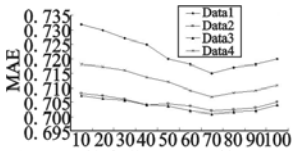

图1　θ的变化对MAE的影响

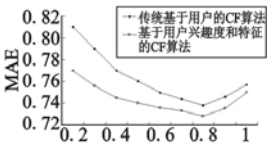

图2　训练集所占比例

从实验结果可以看出,随着训练集数据的增多,效果接近原算法,当训练集所占比例为 80% 时,本文算法的 MAE 达到最低值,算法达到最优。

实验 3　本实验测试当用户邻居数目不同时,本文算法与传统算法的比较。实验结果如图 3 所示。从图 3 中可以看出,当邻居数较大时,准确率更明显,MAE 值更小。但用户最近邻居数目的选取对算法准确率有很大的影响,过大或过小都会产生负面影响。本实验中,邻居数取 3040 间比较合适。由于本文算法对用户特征进行了分析,对同一个项目感兴趣的用户的特征会比较相似,那么根据贝叶斯算法就可以计算出具有相似特征的用户对此项目的喜欢概率,在用户相似度的计算中加入此概率值,使得同时喜欢项目 i 的两个用户或者同时不喜欢项目 i 的两个用户具有更高的相似度。

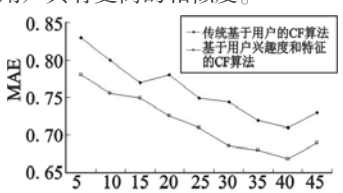

图3　最近邻居数目

综合以上三组实验可以看出,本文算法比传统的基于用户的 CF 算法的推荐质量要好,尤其是共同评分阈值的加入、用户兴趣度以及用户特征分析的引入对目标用户的推荐质量有了很大的改善。

该算法的主要目的是提高用户最近邻居集合的有效性和准确度,从而提高推荐质量。尽管随着用户和项目数目的增加,评分数据增多,数据稀疏性在一定程度上可能有所增加,但是评分数据的增加在一定程度上有利于推荐精度的提高。在文献[4]中提出通过奇异值分解的方法减少项目空间的维数,使用户在降维后的项目空间上对每个项目均有评分。此方法能有效提高推荐系统的伸缩能力,但是降维会导致信息丢失,降低了推荐质量,尤其是项目的空间维数很高时,降维效果难以保证。保留的维数 k 很重要,如果太小,则不能得到原始评分矩阵中的重要结构;如果太大,则失去了降维的意义,所以对保留维数的把握起关键作用。因此,今后的工作将主要研究如何在避免信息丢失的前提下,更有效地解决数据稀疏性问题,优化本文算法,提高算法的可扩展性和推荐质量。

5　结束语

本文将传统的基于用户的协同过滤推荐算法进行优化,首先在用户相似度的计算中引入阈值 θ,对用户相似度进行初始调整;然后利用用户对项目的兴趣度将用户分组,用贝叶斯算法对用户特征进行分析,从而计算出具有不同特征值的用户对未评分项目喜欢的概率;最后与兴趣度结合,确定调节因子 δ 的取值,将用户相似度公式进一步优化,从而使用户最近邻居的计算更加精确。实验结果表明,本文算法可以在一定程度上提高系统的推荐质量。今后的研究方向将是进一步改进算法、提高算法的效率以及可扩展性。

参考文献:

[1] 彭德巍,胡斌.一种基于用户特征和时间的协同过滤算法[J].武汉理工大学学报,2009,31(3):24-28.

[2] 蔡浩,贾宇波,黄成伟.结合用户信任模型的协同过滤推荐方法研究[J].计算机工程与应用,2010,46(35):148-151.

[3] GREINER R, SU Xiao-yuan, SHEN Bin, *et al*. Structural extension to logistic regression: discriminative parameter learning of belief net classifiers[J]. Machine Learning,2005,59(3): 297-322.

[4] 赵亮,胡乃静,张守志. 个性化推荐算法设计[J]. 计算机研究与发展,2002,39(8):986-991.

[5] HAN Peng, XIE Bo, YANG Fan, *et al*. A scalable P2P recommender system based on distributed collaborative filtering[J]. Expert Systems with Applications,2004,27(2): 203-210.

[6] SATO A, ANDO T, INAKOSHI H, *et al*. Personlization system based on dyanamic learning[J]. Journal of the Royal Statistical Society,1997,38(1): 44-59.

[7] 郭艳红,邓贵仕.协同过滤的一种个性化推荐算法研究[J].计算机应用研究,2008,25(1): 39-41,58.

[8] 彭玉. 基于用户个人特征的多内容项目协同过滤推荐[D].重庆:西南大学,2007.

[9] 马宏伟,张光卫,李鹏.协同过滤推荐算法综述[J].小型微型计算机系统,2009,30(7):1282-1288.

[10] LU Hai-ping, PLATANIOTIS K N, VENETSANOPOULOS A N. Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition[J]. IEEE Trans on Neural Networks,2009,20(1):103-123.

[11] 李春,朱珍民,高晓芳,等.基于邻居决策的协同过滤推荐算法[J].计算机工程,2010,36(13):34-36.