结合项目分类和云模型的协同过滤推荐算法

熊忠阳, 刘　芹, 张玉芳
(重庆大学 计算机学院, 重庆 400044)

摘　要: 为了解决用户评分数据稀疏性问题和传统相似性计算方法因严格匹配对象属性而产生的弊端,结合项目分类和云模型提出了一种改进的协同过滤推荐算法。首先,按项目分类得到类别矩阵;然后利用云模型计算类内项目间的相似度并获取具有最高相似度的邻居项目的评分,为类内未评分项目进行预测填充;再利用云模型计算类内用户间的相似度得到用户邻居,最后给出最终的预测评分并产生推荐。实验结果表明,该算法不仅有效地解决了数据稀疏性及传统相似性方法存在的弊端,还提高了用户兴趣及最近邻寻找的准确性;同时,该算法只需计算新增用户或项目所在的类别即可,大大增强了系统的可扩展性。

关键词: 云模型; 项目分类; 协同过滤; 项目相似性; 推荐系统
中图分类号: TP311　　**文献标志码**: A　　**文章编号**: 1001-3695(2012)10-3660-05
doi:10.3969/j.issn.1001-3695.2012.10.015

Collaborative filtering recommendation algorithm based on item classification and cloud model

XIONG Zhong-yang, LIU Qin, ZHANG Yu-fang
(*College of Computer Science*, *Chongqing University*, *Chongqing* 400044, *China*)

Abstract: In order to solve problem of data sparseness in user rating matrix and the drawback of attributes' strictly matching in traditional similarity calculation method, this paper presented an improved collaborative filtering recommendation algorithm by combining the item classification and cloud model. This method firstly utilized the item classification information and cloud model to compute items inner-similarity, and then got the scores from neighbor items which had gotten the highest similarity and used their scores to forecast the unrated inner-class items. Secondly, this method obtained the user's neighbors through inner-class user similarity gained from the cloud model computing, then gave the final forecast grade and carried out the recommendation. Experimental results show that this algorithm is not only an effective solution to data sparseness and the drawbacks of traditional similarity method, but also improves the accuracy of user interest and nearest neighbor search. At the same time, the algorithm that only calculates the categories which adds the new users or items, it greatly increases the scalability of the system.

Key words: cloud mode; item classification; collaborative filtering; item similarity; recommendation systems

0　引言

随着电子商务系统用户数目和项目数据的急剧增加,为使用户在浩瀚的信息海洋中快速找到自己感兴趣的商品,Amazon、阿里巴巴、当当网等很多大型电子商务系统都使用了各种不同形式的个性化推荐系统,主要有两种技术:基于内容的推荐和协同过滤推荐[1,2]。而近年来,协同过滤推荐是应用最成功的推荐技术之一,且被广泛研究和改进。

最近邻协同过滤技术是目前应用最成功的协同过滤技术[3],主要包括用户最近邻推荐和项目最近邻推荐两种[4,5]。用户最近邻推荐是基于当前目标用户评分最相似的邻居用户对目标项目的评分数据产生推荐信息;项目最近邻推荐是基于当前目标用户对目标项目评分最相似的邻居项目的评分数据产生推荐信息。然而由于大型商务站点用户及商品项的数目庞大且不断增加,同时用户给予评分的商品项很少,通常在1%以下[5~9],导致用户—项目评分矩阵的数据稀疏性严重影响了系统的推荐质量,故不同的解决稀疏性问题的方法也随之出现。

传统的解决数据稀疏性的方法主要有:

a)降维方法,通过奇异值分解[10]或压缩数据[8]降低用户评分矩阵维数来解决数据稀疏性问题。

b)仅利用评分填充解决数据稀疏性问题,如评分预测填充推荐[6,7,11~13]、聚类加权评分填充推荐[14]等方法。这类算法虽弥补了数据稀疏性带来的影响,提高了推荐效率,但并没有充分利用项目本身存在的类别更好地获取用户兴趣。

c)结合评分填充和项目类别的方法,如聚类和协同过滤相组合的推荐算法[9]、提出项目综合相似性方法[15]、基于项目分类信息的协同过滤算法[16,17]。这三个算法虽然考虑到利用项目的聚类或分类信息来更准确地获取用户的兴趣,但在计算相似性时仍然采用传统的相似性方法,从而影响了推荐质量。

d)利用隐式评分填充未评分项目的评分,即可以采用文献[18]中的方法分析用户行为,得出用户对未评分项目的

收稿日期: 2012-03-14; **修回日期**: 2012-04-23

作者简介: 熊忠阳(1962-),男,重庆人,教授,博导,主要研究方向为数据挖掘技术与应用、互联网应用关键技术;刘芹(1987-),女,山东潍坊人,硕士研究生,主要研究方向为个性化推荐、云模型(tianshilq@126.com);张玉芳(1965-),女,重庆人,教授,硕导,主要研究方向为数据挖掘与网络入侵检测等.

评分。

为解决传统相似性计算方法因严格匹配对象属性而产生的弊端,李德毅等人提出基于云模型的相似性度量方法。文献[19,20]主要是在知识层面比较用户相似度,利用云模型计算项目(用户)的相似度来获得项目(用户)的最近邻,不仅克服了传统相似度比较方法严格匹配对象属性的不足,并在一定程度上克服了数据稀疏性带来的负面影响。虽然这两种方法都降低了数据稀疏性,提高了推荐质量,但并未考虑到项目的类别信息影响获得用户兴趣的准确度,也未能考虑到针对用户不同的兴趣,兴趣最近邻是不同的,从而导致推荐系统的质量未能得到根本性的提高。针对该问题,本文提出了结合项目分类和云模型的协同过滤改进算法。

1　云模型简介

1.1　云模型的概念

云模型是李德毅院士提出的一种定性定量转换模型[19~22],是云运算、云聚类、云推理、云控制等方法的基础。云模型主要是实现定性概念与其数值表示之间的不确定性转换,其中正向云发生器是由定性概念到定量表示的过程,逆向云发生器则是由定量表示到定性概念的过程。

定义 1　云和云滴。设 U 是一个用精确数值表示的定量论域,C 是 U 上的定性概念,若定量值 $x \in U$,且 x 是定性概念 C 的一次随机实现,x 对 C 的确定度 $u(x) \in [0,1]$ 是有稳定倾向的随机数

$$u: U[0,1] \quad \forall x \in U \quad x \rightarrow u(x)$$

则 x 在论域 U 上的分布称为云(cloud),每一个 x 称为一个云滴(drop)[19,20,23,24]。

云用期望 Ex(excepted value)、熵 En(entropy)和超熵 He(hyper entropy)三个数字特征来整体表征一个概念。期望 Ex 是云滴在论域中最能代表定性概念的点;熵 En 代表定性概念的不确定性,熵值越大说明定性概念的不确定性就越大;超熵 He 是熵的不确定性度量。用三个数字特征表示的定性概念的整体特征记做 $C(\mathrm{Ex},\mathrm{En},\mathrm{He})$,称为云的特征向量。

正向云算法是实现概念空间到数值空间的转换,即将定性概念的整体特征向量 $C(\mathrm{Ex},\mathrm{En},\mathrm{He})$ 变换为数值表示。逆向云算法是实现从定量值到定性概念的转换,将一组定量数值转换为以云的特征向量 $C(\mathrm{Ex},\mathrm{En},\mathrm{He})$ 表示的定性概念。

1.2　基于云模型的相似性度量方法

电子商务系统一般通过用户评分得到用户对项目的满意程度。假设用户的评分采用离散型数值{1,2,3,4,5}表示,其定性概念即为{非常不满意,不满意,一般,满意,非常满意}五个不同的等级。

定义 2　项目评分频度向量。统计多个用户对一个项目的评分情况,记为 $I = \{I_1, I_2, I_3, I_4, I_5\}$,称为项目评分频度向量,$I_1 \sim I_5$ 为用户对项目相应五个等级的评分次数。

根据项目评分频度向量,通过逆向云算法,可以获得由这些评分转换的定性知识,得到项目评分特征向量,记为 $C_I = (\mathrm{Ex},\mathrm{En},\mathrm{He})$。

定义 3　用户评分频度向量。统计一个用户对多个项目的评分情况,记为 $U = \{u_1, u_2, u_3, u_4, u_5\}$,$u_1 \sim u_5$ 分别为用户给出的相应于五个等级的评分次数,称为用户评分频度向量。

根据用户评分频度向量,通过逆向云算法,可以获得由这些评分转换的定性知识,得到用户评分特征向量,记为 $C_U = (\mathrm{Ex},\mathrm{En},\mathrm{He})$。

在定义 2、3 的评分特征向量中,期望 Ex 表示用户对项目的平均满意度,为偏好水平;熵 En 反映了评分的集中程度,为评分的离散度;超熵 He 为熵的稳定度。

定义 4　云的相似度。给定由两个云 i 和 j 的数字特征组成的向量 V_i 和 V_j,它们之间的余弦夹角称为云 i 和 j 之间的相似度:

$$\mathrm{sim}(i,j) = \cos(V_i, V_j) = \frac{V_i \cdot V_j}{\| V_i \| \times \| V_j \|} \tag{1}$$

其中,$V_i = (\mathrm{Ex}_i, \mathrm{En}_i, \mathrm{He}_i)$,$V_j = (\mathrm{Ex}_j, \mathrm{En}_j, \mathrm{He}_j)$。

定义 5　项目相似度。两个项目的评分特征向量之间的夹角余弦,称为它们之间的项目相似度。

定义 6　用户相似度。两个用户的评分特征向量之间的夹角余弦,称为它们之间的用户相似度。

1.3　常用的协同过滤推荐算法优缺点分析

协同过滤算法主要面临三个挑战:数据稀疏性问题、冷启动问题和可扩展性问题。目前大部分研究主要围绕数据稀疏性问题,采用对未评分项目进行评分填充的方法来弥补数据稀疏性的缺陷。对未评分项填充方法中最简单的是缺省值法,即将所有未评分项目的评分填充为评分均值[14]或全部填充为零值[19];但目前用得最多的是评分预测填充方法[12,16,17,20],文献[16,17]不仅考虑到对未评分项目进行评分填充,弥补数据稀疏性,还考虑到结合项目分类信息获得更准确的用户兴趣最近邻。但上述方法在计算项目(用户)相似度时采用的是传统的相似性度量方法,严格匹配对象属性,导致影响了系统的推荐质量,使得推荐效果不理想。

2007 年张光卫等人[19]提出了基于云模型的协同过滤算法,在知识层面比较用户相似度方法,克服了传统相似度计算方法中完全匹配对象属性的缺陷;但该算法将未被用户进行评分的项目的分值全部设为 0,导致用户兴趣最近邻获得不准确。2010 年徐德智等人[20]针对该问题,提出基于云模型的评分预测推荐算法,将用户对指定项目的相似项目的评分经过加权得到用户对指定项目的预测评分,弥补了所有用户对未评分项目都是 0 的缺陷;但该算法没有考虑到项目分类的信息,即针对用户不同的兴趣,用户的兴趣最近邻也应该不同的问题,导致了求目标用户对所有兴趣的兴趣最近邻都是相同的,从而影响了系统的推荐质量。

针对传统相似性度量方法存在的问题及云模型未考虑项目分类信息产生的缺陷,本文提出了一种结合项目分类信息和云模型的协同过滤算法,不仅可以弥补数据稀疏性的不足,还可以只更新项目类别所在的类别矩阵,提高系统的可扩展性。

2　结合项目分类和云模型的协同过滤算法

准确地获取用户的兴趣是个性化推荐服务获得较高推荐质量的重要因素。文献[20]虽然在知识层面获得了比较准确的用户兴趣,但针对目标用户的所有兴趣,其兴趣最近邻都是固定的,这不仅与实际情况不符,而且也影响了系统的推荐质量。故本文结合项目分类和云模型优化了传统的协同过滤算法,针对用户不同的兴趣给予不同的用户兴趣最近邻,从而更好、更准确地获取用户兴趣。

2.1 结合项目分类和云模型的评分预测

在实际的电子商务网站中,任意一个项目都属于一个或多个商品类别,即项目类别矩阵 $C_{I,C}$,其中 I 表示商品项,C 表示类别。假设项目类别 $C = C_1 \cup C_2 \cup \cdots \cup C_j$ 表示所有项目组成的集合,则项目类别矩阵 $C_{I,C}$ 为

$$C_{I,C} = \begin{bmatrix} I_1 & C_{1,1} & \cdots & C_{1,j} \\ I_2 & C_{2,1} & \cdots & C_{2,j} \\ \vdots & \vdots & & \vdots \\ I_i & C_{i,1} & \cdots & C_{i,j} \end{bmatrix}$$

其中:$I_1 \sim I_i$ 表示项目 ID, $C_{i,j}$ 的值只能为 0 或 1,当其值为 1 时,表示第 i 个项目属于第 j 个类别;当其值为 0 时,则表示第 i 个项目不属于第 j 个类别。

用户—项目评分矩阵 $R_{m\times n}$ 是一个 m 行 n 列的矩阵:

$$\begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,n} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,n} \\ \vdots & \vdots & & \vdots \\ R_{m,1} & R_{m,2} & \cdots & R_{m,n} \end{bmatrix}$$

其中:m 行表示用户数;n 列表示项目数;$R_{i,j}$ 表示用户 i 对项目 j 的评分,分值为 1 ~ 5 的整数,数值越大表示用户对项目越喜欢。

对矩阵中 $R_{\mathrm{UID,IID}}$ 分情况讨论:当 $R_{\mathrm{UID,IID}}$ 不为空时,$R_{\mathrm{UID,IID}}$ 为用户 UID 对项目 IID 的评分 $r_{\mathrm{UID,IID}}$;当 $R_{\mathrm{UID,IID}}$ 为空时,则根据 $R_{\mathrm{UID,IID}}$ 项目所在的类别进行类内评分预测。具体算法如下:

算法 1 结合项目分类和云模型的评分预测算法

输入:用户评分表 $R_{m\times n}$ 和项目类别矩阵。

输出:目标用户 UID 对项目 IID 的预测评分 $P_{\mathrm{UID,IID}}$。

1)计算类别用户—项目矩阵

根据用户评分表 $R_{m\times n}$ 和项目类别矩阵 $C_{I,C}$,依次获得每一个类别中的用户—项目评分矩阵。具体方法是:按列值依次读取项目类别矩阵中的第二列数据,当数值为 1 时,获取项目类别矩阵中的 IID(即项目 ID,第一列数据),并根据该 IID 读取用户评分表中对应 IID、UID(用户 ID)及评分记录 rating。当第二列数据读取完毕时便得到所有属于第一类的用户—项目评分矩阵。按照此方法,依次读取项目类别矩阵中的第 2 列、第 3 列、…、第 j 列获得其他类别的用户—项目评分矩阵。

2)计算类内项目评分频度向量

根据 1)获得的类别用户—项目评分矩阵,在每个类内,按照列分别获得类内每个项目的评分频度向量 $I = \{I_1, I_2, I_3, I_4, I_5\}$,$I_1 \sim I_5$ 为所有用户对该项目相应五个等级的评分次数。

3)计算类内项目评分特征向量

根据项目的评分频度向量,通过逆向云算法计算类内每个项目的评分特征向量 $C_{Ii} = (\mathrm{Ex}_i, \mathrm{En}_i, \mathrm{He}_i)$,$i$ 为类内用户数目。

4)计算类内项目相似度矩阵

根据类别划分,采用式(1)分别计算类内项目 i 和项目 j 的相似度 $\mathrm{sim}(i,j)$,得到每个类别的项目相似矩阵 $\mathrm{sim}[I]$(I 为类别 ID,假设类内项目个数为 m):

$$\mathrm{sim}[I] = \begin{bmatrix} \mathrm{sim}(1,1) & \mathrm{sim}(1,2) & \cdots & \mathrm{sim}(1,m) \\ \mathrm{sim}(2,1) & \mathrm{sim}(2,2) & \cdots & \mathrm{sim}(2,m) \\ \vdots & \vdots & & \vdots \\ \mathrm{sim}(m,1) & \mathrm{sim}(m,2) & \cdots & \mathrm{sim}(m,m) \end{bmatrix}$$

其中:$\mathrm{sim}[I]$ 表示第 I 类的用户相似度矩阵,为对称矩阵,总共有 19 个类别,去除类别 0(未知的异常数据),共 18 个类别相似矩阵。

5)获得目标用户 UID 对项目 IID 的预测评分 $P_{\mathrm{UID,IID}}$

a)根据类内项目相似度矩阵,将相似度最高的前 N 项作为 IID 的类内邻居集合 $M_{\mathrm{IID}} = \{I_1, I_2, \cdots, I_N\}$,且 $\mathrm{IID} \notin M_{\mathrm{IID}}$,项目 $I_1, I_2, \cdots, I_N$ 与项目 IID 的相似值依次减小。

b)根据类内邻居集合 M_{IID},采用文献[19]中提出的方法预测用户 UID 对项目 IID 的预测评分:

$$P_{\mathrm{UID,IID}} = \frac{\sum_{n\in M_{\mathrm{IID}}} \mathrm{sim}_{\mathrm{IID},n} \cdot R_{\mathrm{UID},n}}{\sum_{n\in M_{\mathrm{IID}}} |(\mathrm{sim}_{\mathrm{IID},n})|} \tag{2}$$

其中:M_{IID} 是项目 IID 的类内相似项目邻居集合,$\mathrm{sim}_{\mathrm{IID},n}$ 是项目 IID 与类内项目 n 的相似性;$R_{\mathrm{UID},n}$ 是用户 UID 对类内项目 n 的评分。故对任意的项目 $\mathrm{IID} \in U_{\mathrm{UID,IID}}$,用户 UID 对项目 IID 的评分为

$$R_{\mathrm{UID,IID}} = \begin{cases} r_{\mathrm{UID,IID}} & \text{if user UID rated item IID} \\ P_{\mathrm{UID,IID}} & \text{if user UID not rated item IID} \end{cases}$$

2.2 结合项目分类和云模型的协同过滤算法

为获得更准确的用户邻居及推荐结果,本文根据项目分类信息和云模型的相似性度量方法,提出了一种结合项目分类和云模型的协同过滤算法。首先通过项目相似度公式(定义 5)计算类内项目相似度得到项目邻居来预测未评分项目的评分,然后通过用户相似度公式(定义 6)计算类内用户相似性来获得目标用户邻居及推荐值,最后产生推荐。具体算法如下:

算法 2 结合项目分类和云模型的协同过滤算法

输入:用户评分表 $R_{m\times n}$ 和项目类别矩阵 $C_{i,j}$。

输出:目标用户 UID 对项目 IID 的预测评分 $P_{\mathrm{UID,IID}}$

1)获得类内用户—项目矩阵

根据用户评分表 $R_{m\times n}$ 和项目类别矩阵 $C_{i,j}$,计算用户项目矩阵 $R_{m\times n}$。其中,m 行代表用户数,n 列代表项目数。用户 UID 对项目 IID 的评分 $R_{\mathrm{UID,IID}}$ 采用算法 1 得到。

2)计算类内用户评分频度向量

根据 1)得到的类内用户—项目矩阵,分别统计各个类别中用户的评分频度向量 $U_i(u_1, u_2, u_3, u_4, u_5)$($1 \leqslant i \leqslant m$)($m$ 为类别内用户数目),然后根据逆向云算法计算类内每个用户的评分特征向量 $C_{ui} = (\mathrm{Ex}_i, \mathrm{En}_i, \mathrm{He}_i)$,$1 \leqslant i \leqslant m$。

3)计算类内用户相似度矩阵

根据类别划分,采用式(1)分别计算类内用户 C_i 和用户 UID 的相似度 $\mathrm{sim}(C_i, \mathrm{UID})$,得到每个类别的用户相似矩阵 $\mathrm{sim}[i]$($1 \leqslant i \leqslant 18$):

$$\mathrm{sim}[i] = \begin{bmatrix} \mathrm{sim}(1,1) & \mathrm{sim}(1,2) & \cdots & \mathrm{sim}(1,m) \\ \mathrm{sim}(2,1) & \mathrm{sim}(2,2) & \cdots & \mathrm{sim}(2,m) \\ \vdots & \vdots & & \vdots \\ \mathrm{sim}(m,1) & \mathrm{sim}(m,2) & \cdots & \mathrm{sim}(m,m) \end{bmatrix}$$

其中:$\mathrm{sim}[i]$($1 \leqslant i \leqslant 18$)表示第 i 类的用户相似度矩阵,为对称矩阵,实验使用的数据集中总共有 19 个类别,去除类别 0(未知的),共 18 个类别相似矩阵。

4)产生推荐值

将目标项目 IID 所在的类内相似度最高的若干项作为目标用户 UID 的邻居用户,即在类内寻找用户 UID 的最近邻集合 $C_{\mathrm{UID}} = \{C_1, C_2, \cdots, C_k\}$,其中用户 C_1 与用户 UID 的相似度最高,C_2 次之,依此类推。

根据最近邻居集合 C_{UID} 中每个用户对项目 IID 评分的加权平均值,得到 $P_{\mathrm{UID,IID}}$,计算方法如下:

$$P_{\mathrm{UID,IID}}=\overline{R_{\mathrm{UID}}}+\frac{\sum_{u\in C_{\mathrm{UID}}}\mathrm{sim}(\mathrm{UID},u)\times(R_{u,\mathrm{IID}}-\overline{R_u})}{\sum_{u\in C_{\mathrm{UID}}}|\mathrm{sim}(\mathrm{UID},u)|} \tag{3}$$

其中:$R_{u,\mathrm{IID}}$为用户 u 对项目 IID 的评分;sim(UID,u)为用户 UID 与用户 u 的相似度;$\overline{R_{\mathrm{UID}}}$、$\overline{R_u}$分别表示用户 UID 和用户 u 对项目的平均评分。

5)作出推荐

当目标项目 IID 属于多个类别时,则按照式(3)分别计算每个类内的推荐值,然后将所有类的推荐值取平均值,再做"四舍五入"[3]作为最终推荐值 $P_{\mathrm{UID,IID}}$。

3　实验结果及分析

3.1　实验数据集及评估指标

实验采用的数据集是 Movielens 站点(http://movielens. umn. edu)提供的 100k 的公开数据集(1997 年 9 月 19 日—1998 年 4 月 22 日的数据集)。该数据集包括 943 个用户对 1 682 部电影的评分记录。该站点要求注册用户必须至少对它所拥有的电影中的 20 部进行评价才可以使用该系统,故每个用户至少对 20 部电影进行过评价。同时该数据集包含 19 个不同的电影类别,每一部电影至少属于一个类别且可以同时属于多个类别,实验时只利用 1 ~ 18 类进行测试(0 类为未知类,少数异常数据,故不用)。用户评分的范围是 1 ~ 5,评分数值越大表示用户对电影的兴趣越大,即 1 表示最不喜欢,5 表示最喜欢。用户评分数据集的稀疏等级[19]为 1 - 100 000/(943 × 1 682) = 0. 937 0。

本文将数据集按照 80% 和 20% 的比例分为训练集和测试集两部分,取该比例是为了与传统的基于项目分类的协同过滤算法(T-IC CF)[17]、基于云模型的协同过滤算法(LICM CF)[19]、基于云模型的项目评分预测推荐算法(C-IRP CF)[20]进行比较。

本文采用目前使用最广泛的评价推荐系统精确度的标准——平均绝对偏差(mean absolute error,MAE)来衡量推荐质量的好坏。MAE 主要是计算系统预测推荐值与用户真正评价值之间的偏离程度,MAE 值越小,系统的推荐质量越高。

假设预测的用户评价集合表示为$\{p_1,p_2,\cdots,p_N\}$,而相应的实际的用户评价集合为$\{r_1,r_2,\cdots,r_N\}$,则 MAE 的值可采用式(4)计算:

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|p_i-r_i|}{N} \tag{4}$$

3.2　实验内容及结果分析

实验 1　观察本文算法参数取值不同时结果的变化趋势

a)当项目邻居不变时,用户邻居从 10 增加到 50 的 MAE 值变化情况;

b)对比项目邻居个数从 10 增加到 100 的 MAE 值变化情况。

实验结果如图 1 所示。当用户的最近邻固定时,系统的推荐精度随项目邻居的增加而提高;当项目邻居固定时,新算法在用户邻居为 50 时取得最优推荐。用户邻居 userNum = 50,itemNum = 100 时,本算法表现出最好的推荐效果,MAE = 0. 7302。

实验 2　比较本文算法与 T-IC CF[17]算法

在同等数据集及相同比例训练集和测试集的基础上,比较本文算法与 T-IC CF 算法项目邻居个数分别为 50 和 100 时的 MAE 值。图中 C 表示采用云模型(cloud)的分类算法,即本文算法;T 表示采用传统的(traditional)分类算法,即 T-IC CF 算法算法。

实验结果如图 2 所示。当利用项目分类时,采用传统的相似度计算方法,在项目邻居个数比较少时,因严格匹配对象属性获得更好的推荐效果;但当项目邻居个数比较多时,采用云模型的相似度计算方法能获得更准确的推荐效果。

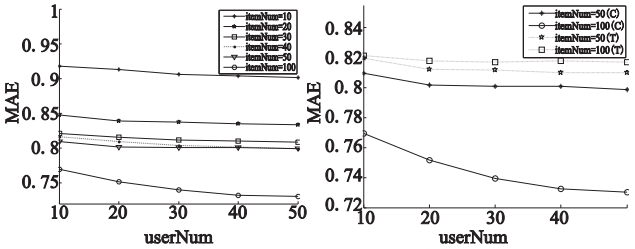

图1　不同邻居的协同过滤算法比较

图2　本文算法与T-IC Cf算法比较

实验 3　观察本文算法与对比算法的推荐效果

在同等数据集及相同比例训练集和测试集的基础上,比较本文算法、T-IC CF 算法[17]与 LICM CF 算法[19]的推荐效果。

实验结果如图 3 所示。用户邻居确定的情况下,三种算法的 MAE 值都随着项目邻居的增加而变小,并且本文算法 MAE 值从 0. 7695 变化到 0. 7302,较其他两种算法有更好的推荐效果;在项目邻居固定的情况下,三种算法的 MAE 值都随着用户邻居的增加而变小,用户邻居超过 50 时,三种算法的 MAE 值变化趋于稳定。

实验 4　对比本文算法与文献[20]的推荐效果

在相同比例训练集和测试集的基础上,比较本文算法与文献[20]的推荐效果。实验结果如图 4 所示。

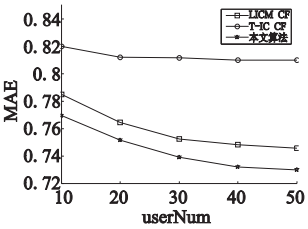

图3　推荐算法的MAE比较

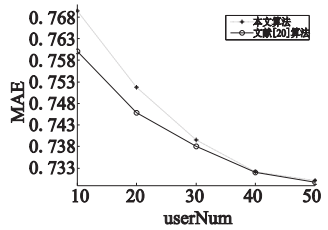

图4　对比本文算法与文献[20]算法

在 useNum = 50,itemNum = 100 的情况下,本文算法和文献[20]算法都得到了最小的 MAE 值。但本文算法的最优 MAE 值 0. 7302 要大于文献[20]的最优 MAE 值 0. 726。分析原因可以发现,两篇论文采用的数据集不同,数据集中的实验数据也略有差距,从而导致实验结果略有偏差,在误差允许范围内属于正常实验结果。

3.3　实验结果分析

综上分析,针对用户的不同兴趣,寻找的最近邻用户或项目的最近邻也不同,寻找的最近邻居更准确。通过实验验证,该方法能更准确地获得用户的兴趣最近邻,提高系统的推荐质量。另外还发现,本文提出的算法仅仅计算用户兴趣增加所在的类即可,而非计算全部数据,提高了系统效率及可扩展性。

4　结束语

针对电子商务系统中存在的数据稀疏性和传统计算相似性方法存在的弊端,本文采用结合项目分类和云模型的协同过

滤算法,可以更准确地获取用户的兴趣,提高了推荐质量。实验结果表明,该算法具有较小的 MAE 值和较高的推荐质量。今后研究中,将重点考虑如何利用用户的分类信息进一步地提高系统推荐质量的问题。

参考文献:

[1] WANG Yi-fan, CHUANG Yu-liang, HSU M H, *et al*. A persona-lized recommender system for the cosmetic business[J]. Expert Systems with Applications,2004,26(3):427-434.

[2] PAPAGELIS M, PLEXOUSAKIS D. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents [J]. Engineering Applications of Artificial Intelligence,2005,18 (7):781-789.

[3] 李春, 朱珍敏, 高晓芳. 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.

[4] GONG Song-jie, YE Hong-wu. Joining user clustering and item based collaborative filtering in personalized recommendation services[C]// Proc of International Conference on Industrial and Information Systems. Washington DC: IEEE Computer Society,2009:149-151.

[5] 黄裕洋, 金远平. 一种综合用户和项目因素的协同过滤推荐算法[J]. 东南大学学报:自然科学版,2010,40(5):917-921.

[6] 李聪,梁昌勇,马丽. 基于邻域最近邻的协同过滤推荐算法[J]. 计算机研究与发展,2008,45(9):1532-1538.

[7] 马丽. 基于组合加权评分的 item-based 协同过滤算法[J]. 情报分析与研究, 2008(11): 60-64.

[8] 侯翠琴,焦李成,张文革. 一种压缩稀疏用户评分矩阵的协同过滤算法[J]. 西安电子科技大学学报:自然科学版,2009,36(4): 614-618.

[9] 刘旭东,葛俊杰,陈德人. 一种基于聚类和协同过滤的组合推荐算法[J]. 计算机工程与科学,2010,32(12):125-133.

[10] SARWAR B M, KARYPIS G, KONSTAN J A, *et al*. Application of dimensionality reduction in recommender system: a case study[C]// Proc of ACM WebKDD Workshop. 2000.

[11] 汪静,印鉴,邓利荣,等. 基于共同评分和相似性权重的协同过滤推荐算法[J]. 计算机科学,2010,37(2):99-104.

[12] 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤算法推荐算法[J]. 软件学报, 2003,14(9):1621-1628.

[13] 周军锋,汤显,郭景峰. 一种优化的协同过滤推荐算法[J]. 计算机研究与发展,2004,41(10):1842-1847.

[14] 傅鹤岗,王竹伟. 对基于项目的协同过滤推荐系统的改进[J]. 重庆理工大学学报:自然科学,2010,24(9):69-74.

[15] 李聪,梁昌勇,董珂. 基于项目类别相似性的协同过滤推荐算法[J]. 合肥工业大学学报:自然科学版,2008,31(3):360-363.

[16] 张海鹏,离烈彪,李仙,等. 基于项目分类预测的协同过滤推荐算法[J]. 情报学报,2009,19(6):218-223.

[17] 熊忠阳,刘芹,张玉芳,等. 基于项目分类的协同过滤改进算法[J]. 计算机应用研究,2012,29(2):493-496.

[18] ZHENG Ling, CUI Shuo, YUE Dong, *et al*. User interest modeling based on browsing behavior[C]//Proc of the 3rd International Conference on Advanced Computer Theory and Engineering. 2010:455-458.

[19] 张光卫,李德毅,李鹏,等. 基于云模型的协同过滤推荐算法[J]. 软件学报,2007,18(10):2403-2411.

[21] CHENG Xi-jun, CHEN Juan-juan. Modeling user interests based on cloud model for masquerade detection[C]//Proc of International Conference on Computational Intelligence and Software Engineering. 2009:668-1336.

[20] 徐德智,李小慧. 基于云模型的项目评分预测推荐算法[J]. 计算机工程,2010,36(17):48-50.

[22] CHENG Xi-jun, CHEN Juan-juan. Evaluation of user interests with a cloud model[C]//Proc of International Conference on Computational Intelligence and Software Engineering. 2009:1-5.

[23] 李德毅,杜鹢. 不确定性人工智能[M]. 北京:国防工业出版社, 2005,143-185.

[24] WANG Bing, TAO Zhao-wen, HU Jun. Improving the diversity of user based top-*N* recommendation by cloud model[C]//Proc of the 5th International Conference on Computer Science & Education. 2010:24-27.