文章编号: 1003-0077(2008)04-0061-05

双向聚类迭代的协同过滤推荐算法

王明文[1],陶红亮[1],熊小勇[2]

(1. 江西师范大学 计算机信息工程学院,江西 南昌 330022;
2. 江西师范大学 软件学院,江西 南昌 330022)

摘 要: 协同过滤是电子商务推荐系统中广泛采用的技术,然而数据稀疏性会影响协同过滤的推荐质量。针对数据稀疏问题提出一种双向聚类迭代的协同过滤推荐算法,对初始得到的用户聚类和项目聚类进行交叉迭代调整,使得聚类簇达到较为稳定的状态。调整后聚类簇的内聚性更强,类之间的区分度更大。实验表明,在调整后的聚类簇中查找邻居将更加准确,可以有效解决数据稀疏问题的影响,有利于提高推荐的准确性。

关键词: 计算机应用;中文信息处理;协同过滤; 聚类; 交叉迭代; 平均绝对偏差
中图分类号: TP391 **文献标识码**: A

A Collaborative Filtering Recommendation Algorithm Based on Iterative Bidirectional Clustering

WANG Ming-wen[1], TAO Hong-liang[1], XIONG Xiao-yong[2]

(1. School of Computer Information and Engineering, Jiangxi Normal University, Nanchang, Jiangxi 330022, China;
2. School of Software, Jiangxi Normal University, Nanchang, Jiangxi 330022, China)

Abstract: Collaborative filtering is widely applied in E-Commerce recommendation system. However, data sparcity affects the accuracy of prediction and results in poor recommendation. To address this problem, a novel collaborative filtering algorithm is presented based on the iterative bidirectional clustering method. It works on the initial user clusters and the item clusters, adjusting the two groups of clusters into the stable status by the cross iteration so that the distances within the cluster are much smaller whereas the distances between the clusters are even bigger. The experiments illustrate that the adjusted clusters facilitate a more accurate neighbor search, indicating an efficient solution to the data sparcity and better recommendation quality.

Key words: computer application; Chinese information processing; collaborative filtering; clustering; cross iteration; mean absolute error

1 引言

随着 Internet 的普及和电子商务的迅速发展,网上交易的购物方式正在逐渐改变传统的商业经营模式,它极大地方便了人们的工作和生活。但是大型电子商务系统中,商品数以万计,面对数量众多的商品信息,人们往往无法迅速准确地找到自己所需。在这种背景下,推荐系统应运而生,它是根据用户的兴趣爱好,推荐符合用户兴趣的项目或信息,是一种个性化服务系统。推荐系统作为电子商务中的重要技术之一,正得到越来越广泛的研究和关注,具有良好的发展前景和应用价值。

最近邻居协同过滤是目前主要采用的推荐技术,它是基于最近邻居的评分数据对目标用户产生推荐[1]。基于项目聚类的协同过滤是将项目进行聚

收稿日期: 2007-05-31 **定稿日期**: 2008-01-05
基金项目: 国家自然科学基金资助项目(60663007);江西省科技攻关项目(2006-184);江西省教育厅科技项目(2007-129)
作者简介: 王明文(1965—),男,教授,博导,主要研究方向为信息检索、数据挖掘、并行计算;陶红亮(1982—),男,硕士生,主要研究方向为数据挖掘、信息检索;熊小勇(1978—),男,助教,主要研究方向为数据挖掘。

类,在与目标项目最相似的前若干个聚类簇中搜索它的最近邻居[2]。因此目前对协同过滤中应用聚类方法的研究比较多,通常采用的聚类算法有 K-means 聚类、ROCK 聚类、模糊聚类[3]等。但是在大型电子商务系统中,有效的用户评分信息非常有限,推荐系统很难根据这些稀疏的数据准确找到目标项目的最近邻居。文献[4]中根据聚类信息进行平滑处理,对用户未评分的项目赋予初始的平滑值,一定程度上解决数据稀疏的影响,提高推荐精度。文献[5]提出用奇异值分解(SVD)降低项目空间维数,使得用户项目矩阵变得稠密,可以有效解决稀疏问题。但是降维处理会导致信息丢失,在项目空间维度很高的情况下,降维效果难以得到保证,而且矩阵的奇异值分解对数据变化比较敏感,同时缺乏先验信息,使得它的应用受到一定的限制[6]。本文提出双向聚类迭代的协同过滤推荐算法,对用户聚类和项目聚类进行交叉迭代调整,使得聚类簇的内聚性更强,类之间的区分度更大。实验结果表明,在调整后的聚类簇中查找目标项目的邻居更准确,可以有效地提高评分预测的准确度,改善推荐质量。

2 交叉迭代算法

给定用户集合 $U=\{u_1, u_2, \cdots, u_m\}$ 和项目集合 $I=\{I_1, I_2, \cdots, I_m\}$,协同过滤中的用户评分数据,可以用一个用户—项目矩阵 R 来表示,其中 m 行表示 m 个用户,n 列表示 n 个项目,矩阵的每个元素 R_{ij} 代表用户 u_i 对项目 I_j 的评分。

2.1 相似性度量方法

度量项目之间相似性主要有两种方法[7]:

· 余弦相似性 将项目评分看成 m 维的向量,那么项目之间的相似性通过向量之间的夹角余弦值来度量。项目 i 和 j 之间的相似性为:

$$sim(i,j) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \cdot \|\vec{j}\|} \quad (1)$$

· 相关相似性 对项目 i 和 j 共同评分过的用户集合记为 U_{ij},它们之间的相似性 $sim(i,j)$ 通过 Pearson 相关系数(PCC)来度量:

$$sim(i,j) = \frac{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_i) \cdot (R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{ij}} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_{ij}} (R_{u,j} - \bar{R}_j)^2}} \quad (2)$$

$R_{u,i}$ 表示用户 u 对项目 i 的评分,$\bar{R}_i$ 和 $\bar{R}_j$ 表示对项目 i 和 j 的平均评分。

余弦相似性实现起来比较简单,也能较好地度量项目间的相似性,而且计算速度较快,但是在评分数据极端稀疏的情况下,通过余弦相似性寻找的邻居不够准确;PCC 相似性考虑了项目的平均评分,可以较好地保证寻找邻居的准确性。本文中,我们分别采用余弦相似性和 PCC 相似性,进行两组实验。

2.2 交叉迭代算法

文献[8]中利用二部图(Bipartite Graph)的概念来描述查询和文档的关系,通过迭代方法计算查询和文档之间的潜在关系,来改善检索效果。同样地在推荐系统中,我们可以认为所有的用户和项目之间总是存在某种关联,有的是用户以显式的方式给出对项目的评分,而另一些是推荐系统预测出用户对其他一些项目的评分[9]。因此也可以用二分图来描述用户和项目的这种关联关系,但是由于某些用户对一些项目有相似的评分,可以认为他们是邻居,因此与一般的二分图不同,用户之间、项目之间是相互关联而不是孤立的,将用户和项目分别进行聚类可以较为快速地找到其邻居。本文采用 K-means 聚类法,对用户和项目分别进行聚类。由于用户—项目矩阵的稀疏性,使得初始得到的聚类不够精确。因此我们采用交叉迭代法,将用户聚类和项目聚类相互调整。

交叉迭代算法的基本思想是:在初始聚类基础之上,计算聚类簇中各个元素与类中心的关联关系,大于某一阈值的保留在该聚类簇中,否则就从当前聚类簇中分离出去,并重新计算该元素与其他聚类中心的相似性,将其加入到与它相似性最大的一个聚类簇中。每次迭代可以将元素划分到与它更相似的聚类簇中,同时聚类簇也是不断地在向比较稳定的状态调整。迭代过程进行一定的次数,当各个聚类簇比较稳定时停止,以此达到调整聚类簇的目的。调整之后的聚类簇内聚性更强,类间区分度更大,在此基础上再进行邻居查找,将会有效地提高推荐的准确性。交叉迭代的过程类似于强化学习(Reinforce Learning)的过程,聚类簇相互进行学习,以调整自身的元素组成。

我们给出交叉迭代调整的具体步骤如下:

1) 用户聚类调整

利用项目聚类信息,来调整用户聚类。在各个用户聚类簇中,通过下式计算每个用户 u_t 与类中心 u_c 的关联关系。

$$S_u(u_t, u_c) = \begin{cases} 1 & u_t = u_c \\ \dfrac{1}{\sum_{k=1}^{n} r_{tk} \sum_{k=1}^{n} r_{ck}} \sum_{I_i \in CommItem} \sum_{I_j \in CommItem} Sim(I_i, I_j) & else \end{cases} \quad (3)$$

其中, r_{tk} $(k = 1, 2, \cdots, n)$ 是 u_t 对各个项目的评分, r_{ck} $(k = 1, 2, \cdots, n)$ 是 u_c 对各个项目的评分。$CommItem$ 是 u_t 和 u_c 共同评分过的项目集合。如果 u_t 和 u_c 共同评分过的项目比较多,可以认为他们是兴趣相似的用户,(3) 式可以很好地体现这个思想。如果得到的 $S_u(u_t, u_c)$ 大于某一阈值,就把 u_t 保留在该聚类簇中,否则将 u_t 从当前聚类簇中分离出去,并计算 u_t 与其他聚类中心的相似性,将其加入到相似性最大的聚类簇中,完成对用户聚类的调整。

2) 项目聚类调整

利用用户聚类信息,来调整项目聚类。与用户聚类调整类似,通过下式计算各个项目 I_q 与类中心 I_c 的关联关系,

$$S_I(I_q, I_c) = \begin{cases} 1 & I_q = I_c \\ \dfrac{1}{\sum_{k=1}^{m} r_{kq} \sum_{k=1}^{m} r_{kc}} \sum_{u_i \in CommUser} \sum_{u_j \in CommUser} Sim(u_i, u_j) & else \end{cases} \quad (4)$$

其中, r_{kq} $(k = 1, 2, \cdots, m)$ 是第 k 个用户对项目 I_q 的评分, r_{kc} $(k = 1, 2, \cdots, m)$ 是第 k 个用户对 I_c 的评分。$CommUser$ 是对 I_q 和 I_c 都有评分的用户集合,如果 $CommUser$ 中的用户比较多,那么我们认为 I_q 和 I_c 是相似的项目。同样地,如果 $S_I(I_q, I_c)$ 大于某一阈值,就把 I_q 保留在该聚类簇中,否则将 I_q 从当前的聚类簇中分离出去,并计算 I_q 与其他聚类中心的相似性,将其加入到相似性最大的聚类簇中,完成对项目聚类的调整。具体的交叉迭代算法 (Crossing Iterative Algorithm) 如下:

输入: 用户集合 U,项目集合 I,用户—项目评分矩阵 R
输出: 调整后的用户聚类
K-means 聚类算法,得到初始的用户聚类 $UC = \{uc_1, uc_2, \cdots, uc_k\}$ 和聚类中心 $CC^u = \{cc_1^u, cc_2^u, \cdots, cc_k^u\}$,初始的项目聚类 $IC = \{Ic_1, Ic_2, \cdots, Ic_k\}$ 和聚类中心 $CC^I = \{cc_1^I, cc_2^I, \cdots, cc_k^I\}$;
Repeat
1) //用户聚类调整
For each user cluster $uc_i \in UC$
For each user $u_j \in uc_i$
计算 u_j 与聚类中心 cc_i^u 的关联关系 $S_U(u_j, cc_i^u)$;
If $(S_U(u_j, cc_i^u) > \varepsilon)$
//ε 是某一给定的阈值
continue;
else
计算 u_j 与其他聚类中心的相似性,将其加入相似度最大的聚类簇中;
Endfor
Endfor
重新计算 UC 中的各个聚类的中心,并更新到 CC^u 之中;
2) //项目聚类调整
For each item cluster $Ic_i \in IC$
For each item $I_j \in Ic_i$
计算 I_j 与聚类中心 cc_i^I 的关联关系 $S_I(I_j, cc_i^I)$;
If $(S_I(I_j, cc_i^I) > \eta)$
//η 是某一给定的阈值
continue;
else
计算 I_j 与其他聚类中心的相似性,将其加入相似度最大的聚类簇中;
Endfor
Endfor
重新计算 IC 中的各个聚类的中心,并更新到 CC^I 之中
Until 聚类簇中元素不再分离或者达到设定的迭代次数
迭代算法完成

理论上,当各个聚类簇达到比较稳定时 (即聚类簇中的大部分元素不再分离出去),迭代过程就停止。但是在实际的运算过程中,很难确定聚类簇何时稳定,可行的做法是确定一个迭代的次数,来终止迭代过程。本文后面的实验中,我们设定不同的迭代次数,观察平均绝对偏差 (Mean Absolute Error, MAE) 的变化,实验结果表明,迭代 6 次后 MAE 值基本稳定,即聚类簇达到比较稳定的状态。

2.3 邻居查找和预测推荐

由聚类的性质可知,目标项目的最近邻居大部分都在与它相似性最高的前若干个聚类中,只要在与它相似性最高的前若干个聚类中就能查找到大部分的最近邻居。因此,基于项目聚类的方法可以大大提高在线的最近邻居查找速度,满足推荐系统的实时性要求。

采用 KNN 方法进行最近邻居查找,假设要查找目标项目 t 的 k 个最近邻居,首先计算 t 与各个项目聚类中心的相似度,选取前若干个与它相似度最高的项目聚类作为查询空间,记为 $SearchSpace(t)$。

对 $SearchSpace(t)$ 中的每个项目 t_i,计算 t 与 t_i 的相似度 $sim(t,t_i)$,选取与 t 相似性最大的前 k 个项目作为 t 的最近邻居。

通过 KNN 方法得到目标项目 t 的最近邻居后,根据用户 u 对邻居的评分信息,可以预测他对目标项目 t 的评分。设 t 的最近邻居集合为 $NN(t)$。则用户 u 对 t 的预测评分可以通过 u 对 $NN(t)$ 中所有项目的评分得到,计算公式如下:

$$P_{u,t} = \overline{R}_t + \frac{\sum_{t_i \in NN(t)} sim(t,t_i) \cdot (R_{u,t_i} - \overline{R}_{t_i})}{\sum_{t_i \in NN(t)} |sim(t,t_i)|} \quad (5)$$

其中,$sim(t,t_i)$ 表示目标项目 t 与最近邻居 t_i 的相似度,R_{u,t_i} 表示用户 u 对项目 t_i 的评分,$\overline{R}_t$,$\overline{R}_{t_i}$ 分别表示对项目 t 和 t_i 的平均评分。

最后,根据上述方法预测的用户 u 对所有目标项目的评分,选取预测评分最高的前 N 个项目(top-N)作为推荐结果返回给用户。

3 实验和结果分析

3.1 数据集和评价标准

本文采用 GroupLens 研究小组(http://www.grouplens.org/)提供的 MovieLens 数据集,其中包含 943 个用户和 1 682 部电影,每个用户至少对 20 部电影有评分,总共 100 000 条评分记录。我们将整个数据集分为训练集和测试集两部分,取 80 %的数据作为训练集,20 %作为测试集。

评价推荐系统推荐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[7]。本文采用平均绝对偏差 MAE,MAE 是一种统计精度度量方法,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,推荐质量越高。假设预测的用户评分集合表示为$\{p_1, p_2, \cdots, p_n\}$,对应的实际用户评分集合为$\{q_1, q_2, \cdots, q_n\}$,则平均绝对偏差 MAE 计算公式为:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \quad (6)$$

3.2 聚类数目的选取

K-means 聚类方法中,聚类数 K 的选取非常关键,我们采用类间距与类内距之比作为聚类准则。

类间距定义为:

$$Out\_distance = \frac{1}{n(n-1)} \sum_{i}^{n} \sum_{j}^{n} |c_i, c_j| \quad (7)$$

其中 n 是聚类数,c_i,c_j 分别表示聚类 C_i,C_j 的类中心,$|c_i, c_j|$ 为两个类中心之间的距离。类间距表示类之间的区分程度,一般是越大越好。

类内距定义为:

$$In\_distance = \frac{1}{n} \sum_{i}^{n} \left(\frac{1}{m_i} \sum_{j}^{m} |x_{ij}, c_i| \right) \quad (8)$$

其中 n 是聚类数,m_i 表示聚类 C_i 中元素个数,c_i 表示类 C_i 的中心,x_{ij} 表示类 C_i 中的第 j 个元素,$|x_{ij}, c_i|$ 为类中元素与类中心的距离。类内距表示类内的凝聚程度,一般是越小越好。

一般地,类间距与类内距之比越大,表示聚类效果越好。相应地,在实验中调整聚类数目,得到不同的类间距与类内距之比,达到较好的聚类效果。聚类数目从 10 增加到 30,间隔为 2,分别采用了余弦相似性和 PCC 相似性,进行了两组实验,计算它们的类间距与类内距之比(见图 1,图 2)。

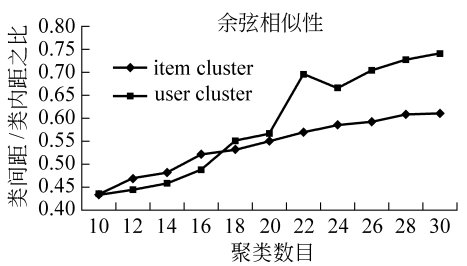

图 1 类间距与类内距之比(余弦相似性)

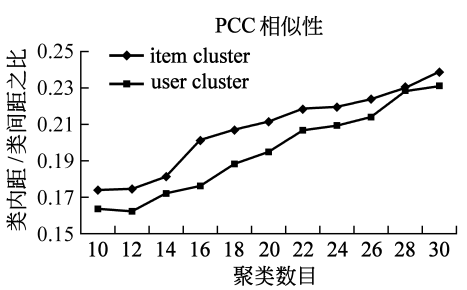

图 2 类间距与类内距之比(PCC 相似性)

图 1 和图 2 中,类间距与类内距之比随着聚类数目的增加而增大。当聚类数为 30 时,类间距与类内距之比最大,而且曲线上升趋势都趋于平缓。如果选取的聚类数大于 30,类间距与类内距之比会更大些,因为从理论上分析,随着聚类数的增加,类间距会增加,类内距会减少,使得它们之比越来越大,但是聚类数目增加的同时带来的是计算量的极剧增长,聚类过程会更加耗时。因此出于算法效率方面

的考虑,在实验中我们选取用户聚类和项目聚类数均为 30。

3.3　实验结果和分析

理论上,当各个聚类簇达到比较稳定时,迭代过程就停止,但是在实际的运算过程中,很难确定聚类簇何时稳定,我们的做法是设定迭代次数,强行终止迭代过程。因此,需要观察不同迭代次数对 MAE 值的影响。下面我们在设定目标项目的邻居数目为 30 的条件下,调整迭代次数从 1～10,观察 MAE 值的变化(见图 3)。

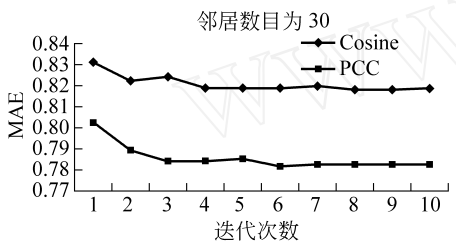

图 3　不同迭代次数时 MAE 的变化曲线图

从图 3 可以看出,迭代 6 次之后,两条 MAE 曲线都趋于直线,我们认为聚类簇达到比较稳定的状态。在迭代调整之后的聚类簇中查找目标项目的最近邻居将更加准确,有利于提高预测评分的准确度。

一般而言,基于项目聚类的方法中扫描整个项目集的 30 %～40 %就可以找到整个项目集的 85 %～90 %的最近邻居[2]。因此,本文实验中最近邻居查询是选取与目标项目最相似的前 8 个项目聚类作为查询空间,邻居数目从 10 增加到 100 个,间隔为 10。实现了三种协同过滤算法: 迭代调整聚类的算法、基于项目聚类的算法和传统的基于项目的算法,分别计算它们的 MAE 值。其中聚类数为 30,迭代次数为 6 次,余弦相似性实验结果如图 4, *PCC* 相似性结果如图 5。

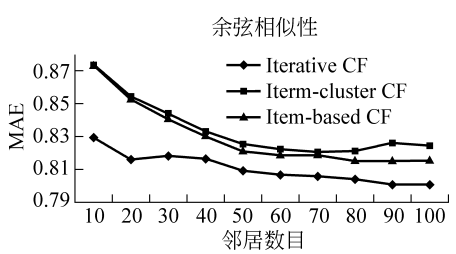

图 4　三种协同过滤算法 MAE 值(采用余弦相似性)

图 4,图 5 中三种算法相比,基于项目(Item-based CF)与基于项目聚类(Item-cluster CF)的 MAE 曲线大致相当,这说明基于项目聚类的方法

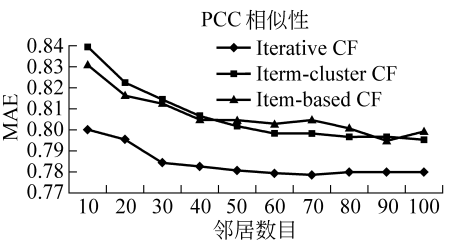

图 5　三种协同过滤算法 MAE 值(采用 PCC 相似性)

在不损失推荐精度的同时,可以大大提高最近邻居查询速度,从而有效地提高推荐系统的在线响应速度,这与文献[2]中的结论是一致的。本文提出的迭代聚类调整算法(Iterative CF)的 MAE 值总体上都比其他两种算法小,推荐精度更高。尤其是相比于 Item-cluster 算法,因为 Item-cluster 算法就是 Iterative CF 迭代 0 次的特殊情况,这说明经过交叉迭代调整之后,在得到的聚类簇中确实可以提高查找邻居的准确度,从而提高推荐精度,有效地解决数据稀疏问题。

一般来讲,大部分成熟的推荐系统的推荐内容本身已有分类体系,那么我们不需要对项目类别做出调整,在这种情况下,迭代算法的作用并不明显。总的来说,本文提出的双向聚类迭代算法更适用于用户类别和项目类别并不明确的系统中。交叉迭代算法的不足之处在于,聚类和交叉迭代的过程提高了算法的时间复杂度,增加了计算量;不过由于电子商务系统中项目的更新相对较为缓慢,因此可以离线对用户和项目进行聚类,迭代调整,将计算结果保存在数据库中,这样能够提高整个电子商务推荐系统的推荐速度和实时响应能力。

4　结束语

随着电子商务规模的日益增长,用户和项目数量急剧增加,推荐系统的实时性、数据稀疏性和可扩展性问题也随之加剧[10]。本文提出双向聚类迭代的协同过滤算法,通过调整用户聚类和项目聚类,可以在一定程度上解决数据稀疏问题,提高推荐质量。

由于 K-means 聚类法中,初始设定的 K 值对聚类效果的影响非常大,因此合理地选取 K 值,是 K-means 聚类法的关键。一般在推荐系统中,用户和项目的数量比较大,如何合理地确定聚类的数目和聚类终止条件,是值得我们进一步研究的问题。另外,K-means聚类法适用于一些数据规模较小的环

(下转第 74 页)

Conference “Recherche d’Information Assistee par Ordinateur”, Montreal: CA,1997: 200-214.

[14] Munirathnam Srikanth and Rohini Srihari, Exploiting Syntactic Structure of Queries in a Language Modeling Approach to IR [C]// Proceedings of CIKM ’03, 2003: 476-483.

[15] Shuang Liu, Fang Liu, Clement Yu and Weiyi Meng, An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases [C]// Proceedings of SIGIR ’04, 2004: 266-272.

[16] Strzalkowski, T., Perez-Carballo, J., Karlgren, J., Hulth, A., Tapanainen, P. and Lahtinen, T. (1999a), Natural Language Information Retrieval: TREC-8 Report [C]// Proceedings of TREC-8, Gaithersburg, Maryland, USA.: 1999.

[17] Strzalkowski, T., Lin, F., Wang, J. and Perez-Carballo, J. (1999b), Evaluating Natural Language Processing Techniques in Information Retrieval [C]// Tomek Strzalkowski (ed.), Natural Language Information Retrieval. Kluwer Academic Publishers, Dordrecht.

[18] Tao Tao, ChengXiang Zhai, An exploration of proximity measures in information retrieval [C]// Proceedings of SIGIR ’07 2007: 295-302.

[19] Thorsten Brants. Natural Language Processing in Information Retrieval [C]// Proceedings of 20th International Conference on Computational Linguistics, Antwerp, Belgium: 2004:1-13.

[20] **王灿辉,张敏,马少平. 自然语言处理在信息检索中的应用综述**[J]. **中文信息学报**,2007,21(2): 35-45.

[21] **赵军,金千里,徐波. 面向文本检索的语义计算**[J]. **计算机学报**,2005,28(12):2068-2078.

(上接第 65 页)

境,而实际的电子商务系统中用户和项目数以万计,这就要求推荐系统有很好的扩展性。因此我们需要考虑如何改进 K-means 聚类法或引入其他的聚类方法,以适应更复杂的环境。

参考文献:

[1] Breese J. S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. 1998. 43-52.

[2] **邓爱林,左子叶,朱扬勇. 基于项目聚类的协同过滤推荐算法**[J]. **小型微型计算机系统**, 2004, 25(9): 1665-1670.

[3] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms [C]// Proceedings of the Tenth International World Wide Web Conference, 2001, 285-295.

[4] Gui-Rong Xue, Chenxi Lin, Qiang Yang. Scalable Collaborative Filtering Using Cluster-based Smoothing [C]// Proceeding of the 28th Annual International ACM SIGIR Conference, in Salvador, Brazil, 2005.

[5] Sarwar B, Karypis G, Konstan J, Riedl J. Application of dimensionality reduction in recommender systems: A case study [C]// ACM WebKDD Web Mining for E-Commerce Workshop, 2000.

[6] Ungar L. H, Foster D. P. Clustering Methods for Collaborative Filtering [C]// Workshop on Recommender Systems at the 15th National Conference on On Artificial Intelligence. 1998.

[7] Aggarwal C C. On the effects of dimensionality reduction on high dimensional similarity search [C]// Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART. Symposium on Principles of Database Systems, 2001, 256-266.

[8] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms [C]// Proceedings of the Tenth International World Wide Web Conference, 2001, 285-295.

[9] Gui-Rong Xue, Hua-Jun Zeng. Optimizing Web Search Using Web Click-through Data [C]// CIKM ’04, November 8-13, Washington, DC, USA: 2004.

[10] **陶红亮. 双向聚类迭代的协同过滤推荐算法**[D]. **南昌: 江西师范大学**, 2007.