信息系统学报
第2卷第2辑(总第3辑)：78－96

China Journal of Information Systems
Vol.2 No.2, Oct. 2008：78－96

基于高维稀疏数据聚类的协同过滤推荐算法*

姚忠，魏佳，吴跃

（北京航空航天大学经济管理学院信息管理与信息系统系，北京 100083）

摘　要　针对协同过滤推荐算法面临数据高维稀疏特征时推荐效果较差的缺点，在现有高维稀疏数据聚类研究的基础上，利用评分数据稀疏差异度和项目类别构造集合差异度度量公式，用以在用户—项目评分矩阵上进行项目聚类。在此基础上进行项目相似性计算和最近邻居查询，然后对用户未评分的项目进行评分预测，进而产生推荐。实验证明本文提出的基于稀疏差异度和项目类别的项目聚类算法及在此基础上的协同过滤推荐结果优于传统的K-means聚类算法基础上的推荐效果。同全项目集协同过滤推荐相比较，在效率和推荐精度上也表现出一定的优越性。

关键词　推荐系统，协同过滤，项目聚类，项目类别评分，IBCRA

中图分类号　TP311

1　引言

Internet推动了电子商务的飞速发展，网络作为一个全新的销售渠道、采购渠道和客户渠道，越来越受到企业和消费者的重视。实施电子商务系统对企业的服务提出了诸多新要求，包括商品质量的保证、送货及时性、商品选购舒适度、退货便利性等，其中最为突出的一个问题就是商品选购的个性化推荐[1]。推荐系统(Recommender System)是解决信息过载的有效手段，也是电子商务服务商提供个性化服务的重要信息工具。电子商务推荐系统是提供“一对一”个性化服务的一种重要的信息技术，它利用电子商务网站向客户提供商品信息和建议，帮助用户决定应该购买什么产品，模拟销售人员帮助客户完成购买过程[2]。推荐系统和个性化推荐技术研究在国内外逐渐成为研究热点，并被广泛应用。Amazon、CDNOW、eBay、Levis、Moviefinder、Reel等众多国外知名电子商务网站，已经将推荐系统集成到运营系统中[3]。相比而言，国内B2C网站虽然在个性化和自动化推荐方面还存在差距[4]，但随着中国电子商务的蓬勃发展，推荐系统的理论研究正逐步深入，国内网站的推荐策略较原先的分类浏览和基于内容的检索等简单方式也更加智能化，推荐技术正逐步应用到网站中来。

1.1　相关文献

推荐系统包括个性化推荐系统和非个性化推荐系统[5]。非个性化推荐系统向所有用户提供具有相同内容的推荐，如电子商务站点的畅销排行。个性化推荐系统则区分不同用户或用户群，根据他们的偏好定制推荐[6]。非个性化推荐系统原理简单，易于操作，但没有考虑到用户需求的差异性，推荐

* 基金项目：国家自然科学基金(70672020,70521001)。
通信作者：姚忠，北航经管学院，副教授、博士，e-mail：iszhyao@buaa.edu.cn。

质量较差；个性化推荐系统算法和实现相对复杂，但推荐质量高。由此，当前电子商务推荐系统的研究基本上集中在个性化电子商务推荐系统的研究领域。

个性化电子商务推荐系统研究依赖用户在评分体系架构中的显式评分，并以此预测用户未评分项目的评分。其中，个性化推荐中使用的推荐算法是通过用户对项目的评分以及附加信息，对尚未评分项目进行评分预测，将评分最高的项目或项目组推荐给用户。推荐系统的个性化推荐服务，提高了客户对电子商务网站的忠诚度，为企业赢得了更多的发展机会[7]。

用户偏好信息的获取是推荐算法的前提。用户信息的获取主要是通过用户对给定信息的评价，主要包括显式评价和隐式评价两类[8]。显示评价基于用户有意识地表达对项目的认可程度，通常使用特定区间的整数值来表达用户的偏好程度，用户数据库中的信息随着用户不断使用而随时更新。隐式评价不需要用户主动参与，推荐系统通过 Agent、Web 数据挖掘等技术自动跟踪并分析用户浏览记录、购物记录等行为来获取信息。

当前，电子商务推荐系统的研究内容和研究方向主要包括推荐方法研究、实时性研究、推荐质量研究、多种数据多种方法的集成、数据挖掘在推荐系统中的应用、用户隐私保护研究等[9]。其中，推荐算法是电子商务推荐系统的核心，推荐系统其他研究内容绝大多数也是以推荐算法为研究出发点。在推荐算法中，主要的研究方向包括协同过滤推荐、基于内容的推荐、聚类技术、Bayesian 网络技术、关联规则技术、基于图的 Horting 图技术等[10]。本文研究高维稀疏数据情况下的协同过滤推荐算法，因此下面主要介绍与此相关的研究。

协同过滤推荐算法是目前最受欢迎的推荐技术[4,11]。Tapestry 是最早提出的协同过滤推荐系统，目标用户需要明确指出与自己行为比较类似的其他用户[2,12]。协同过滤推荐算法在用户对项目做出评价的基础上，通过用户—项目评价矩阵发现用户的共同兴趣模式，预测用户之间的相似度，从而为目标用户做出个性化的推荐。协同过滤推荐主要有两种方法：基于用户的协同过滤(User-Based Collaborative Filtering)和基于项目的协同过滤(Item-Based Collaborative Filtering)。也有研究者提出将两者相结合的方法。协同过滤推荐算法冷启动问题研究，新项目往往是市场上最流行的商品，但执行协同过滤时却不可能将其作为推荐项目，因为没有人对它进行评比或购买，系统无法提供正确的推荐信息。Schein 等[13]通过贝叶斯分类法来解决新项目推荐问题。欧立奇等[14]利用生成树算法划分项目矩阵并计算项目间相似度，根据用户对已有项目的评分和项目间相似性预测用户对新项目的评分。

随着推荐技术的深入发展，大量文献对协同过滤推荐算法进行了改进。改进的协同过滤推荐算法主要研究内容是对相似性算法的改进。采用传统方法计算不同项目之间的相似性时，由于受到用户评分数据稀疏性的影响，难以保证推荐结果的准确性；由于用户对于所有评分为 0 的项目的喜好程度不可能完全相同，余弦相似性不能有效地在没有经过处理的用户——项目矩阵的基础上度量项目之间的相似性；传统的计算项目之间相似性的时候多数忽略了项目之间本身存在的固有关系—项目类别关系，即同属于某一个类别的项目之间应该有更高的相似性[12]。彭玉等[15]提出了一种基于项目的协同过滤推荐算法，通过计算项目的评分相似性和属性相似性，利用双向蕴涵谓词计算项目的相似性。邢春晓等[16]在相似性计算公式中引入了缩放系数 α，以削弱被访问过很多次的资源在相似度计算中的影响。为提高推荐效果，姚忠等[9]在相似性公式中引入调和参数 α，通过调和参数与项目类别数据的乘积来调节项目间的相似性。

集成语境信息的协同过滤推荐算法。传统的推荐技术都是基于用户×项目的二维空间，仅仅是建立在用户对项目的评价信息上来对未评分项目的评分预测，从而进行个性化推荐，没有考虑另外的语境信息，而这些语境信息在一些应用中可能很重要。所谓语境信息就是指对人的行为或者事件的

发展产生影响的上下文信息或者场景信息,如时间、地点等信息[17]。用户消费的语境信息在很大程度上会影响用户的偏好以及最终的购物决策。

多维技术是从数据挖掘中发展起来的一种推荐技术,它扩展了传统的二维矩阵,引入了语境信息。与传统的二维推荐模型需要用全部数据进行预测不同,多维推荐模型在进行评分预测时,只会用到与用户指定的语境信息相关的那些数据[17]。集成语境信息的技术通过建立集成语境信息的多维评分模型,通过选择最优的语境段,用基于降维的方法将多维的模型降低到传统的两维推荐模型上,并在此基础上用协同过滤推荐算法进行评分的预测以及项目的推荐。

数据稀疏性是造成推荐质量低的主要原因之一。为了提高推荐算法的推荐质量,许多研究人员都试图降低数据稀疏性带来的问题,从不同角度对用户和产品信息进行分析、处理、降低数据的稀疏程度。基于项目的协同过滤推荐、降维法、智能 Agent 方法可以在一定程度上缓解数据稀疏性问题。张峰,常会友[18]使用 BP 神经网络缓解协同过滤推荐算法的稀疏性问题,根据用户评分向量交集大小选择候选最近邻居集,采用 BP 神经网络预测用户对项目的评分。Huang 等[19]将奇异值分解应用到协同过滤推荐算法,通过奇异值分解算法得到低维正交矩阵,较好地解决了数据稀疏性问题,但是推荐的准确性会有一定的下降。在采用神经网络模型进行聚类处理的总思路下,Aggarwal 等[20]通过寻找基于对象属性信息的项目类间隐性联系,化解数据稀疏性对高维数据聚类的影响。孙多等[21]阐述了单值分解、聚类等数据稀疏性解决技术。此外,应用聚类算法是解决用户—项目评分矩阵稀疏性的比较有效的方法。由于聚类算法可用于发现数据库中未知的对象类,对象类的划分是考察个体或数据对象间的相似性,将满足相似性条件的个体或数据对象划分为一组,不满足相似性条件的个体或数据对象划分在不同的组[18]。因此,运用聚类算法将具有相似兴趣爱好的用户分配到相同的聚类中;聚类产生后,根据聚类中其他用户对商品的评价预测目标用户对该商品的评价。该方法存在的最大缺陷是如果目标用户处在聚类的边缘,则对该用户的推荐精度比较低[22]。为解决这一缺陷,O'Connor 等人提出了对项目进行聚类,然后再对应聚类中搜索目标项目的最近邻居[23]。

聚类技术经常同协同过滤推荐算法组合在一起,即基于聚类的协同过滤推荐算法。针对高维聚类协同过滤推荐算法在整个项目空间内搜索目标对象的最近邻居,将产生巨大的计算量,严重影响算法的效率。通过聚类技术,将搜索目标对象最近邻居的范围缩小到与目标对象相似性程度最高的几个聚类,可以有效减少计算量,提高实时响应能力。邓爱林等[7]提出了一种基于项目聚类的协同过滤推荐算法,根据用户对项目评分的相似性进行扩展,生成相应的聚类中心,在此基础上计算目标项目与聚类中心的相似性。张海燕等[24]提出了一种通过模糊聚类的方法将项目属性特征的相似性与基于项目的协同过滤推荐技术相结合的推荐算法,在对项目进行模糊聚类后得到项目在属性特征上的相似关系群。何光辉等[12]提出了一种改进的聚类邻居协同过滤推荐算法,利用聚类邻居方法构造邻居关系。高凤荣等[25]采用聚类方法对稀疏矩阵进行划分,通过对资源评分矩阵的划分,以缩小近邻搜索范围和需要预测的资源数目,减少数据稀疏性。孙多等[21]将 Web 日志挖掘技术引入到协同过滤推荐研究中,通过构建兴趣度对用户进行聚类。

以上文献中没有考虑用户—项目评分矩阵中的稀疏差异度,而这一差异度对提高聚类效果有明显的作用[5]。本文针对推荐系统数据集高维稀疏的特征,将评分数据的稀疏差异度引入到项目聚类算法中,然后将稀疏差异度和项目类别构造集合差异度相结合,对用户—项目评分矩阵进行项目聚类,然后,在聚类后的集合内进行项目相似性计算和最近邻居查询,对用户未评分的项目进行评分预测,进而产生推荐。

全文共分五小节。第一节简要阐述了电子商务推荐系统、推荐算法、聚类算法相关领域的研究现状。第二节具体提出了基于稀疏差异度和项目类别的项目聚类算法(IBCRA 算法)。在分析了算法

的基本思想、项目类别、稀疏差异度等重要概念和构造方法后，详细阐述了基于稀疏差异度和项目类别的项目聚类算法的步骤。然后，在基于稀疏差异度和项目类别的聚类算法的基础上，利用该算法进行项目聚类，在目标项目集中进行项目相似性计算和最近邻居查询，通过对目标用户未知项目的评分预测生成项目推荐列表。第三节设计了本文提出的基于项目聚类的协同过滤推荐算法的实验仿真和测试。实验使用 MovieLens 数据集，以平均绝对误差(MAE)为评价指标，首先使用基于稀疏差异度和项目类别的聚类算法对训练集上进行项目聚类，并在此基础上生成最近邻居列表，在测试集上验证算法的效率。第四节对当前商用的聚类算法进行了实验比较分析。结论部分总结了本文的研究成果、尚存在的不足和今后的研究方向。

2 算法设计

基于项目聚类的协同过滤推荐算法主要分为四个过程：项目聚类算法；项目相似性计算；最近邻居查询；评分预测过程算法。下面按照这四个过程介绍。首先介绍项目聚类算法原理与流程，然后阐述项目相似性计算、最近邻居查询和评分预测三个主要环节。项目相似性计算通过相似性计算公式，在用户—项目评分数据集 R 上计算目标项目与邻居项目的相似性 $\mathrm{sim}(i,j)$。最近邻居查询通过相似性计算结果，搜索与目标项目相似程度最高的若干个邻居项目，构成目标项目的最近邻居项目集合 M_j。最近邻居项目集合 M_j 的长度，即最近邻居项目集合元素个数，是由最近邻居查询个数决定的。最近邻居查询个数 $Knear$ 是算法最重要的输入参数之一，$Knear$ 参数取值不仅关系到推荐结果的准确性，而且影响到推荐算法效率。本文中，除目标项目为孤立点或者其他特殊情况外，目标项目相似性计算及最近邻居查询均在目标项目所属的项目类中进行。最后，给出评分预测过程算法。

2.1 项目聚类算法

IBCRA 项目聚类算法首先对评分数据稀疏差异度和项目类别构造集合差异度的度量指标进行计算，通过集合差异度与阈值的比较，判定两个项目集合是否可以归为一类。算法采用“自底而上”的层次聚类法，将每一个项目作为一个类开始，依照项目次序逐个向上聚集，合并最相似的项目。用户—项目评分矩阵 R 是典型的高属性维稀疏矩阵。高属性维稀疏数据对象间的稀疏相似性，可以通过计算对象间的稀疏特征的差异度来描述，差异度越大，对象越不相似；差异度越小，对象越相似，即差异度反映对象间的相似程度。本文的稀疏差异度计算与吴森等[5]的方法不同点，在于描述评分数据的稀疏差异度是通过比较用户对集合内项目评分数据的分布得到的。比较项目集合 C_i 和项目集合 C_j 的相似程度时，首先在用户集 U 上统计用户 $u_i, i=1,2,\cdots,n$。对项目集合 C_i 内有评分的项目个数，再统计用户 $u_i, i=1,2,\cdots,n$ 对项目集合 C_j 内有评分的项目个数。在用户集 U 中挑选出对项目集合 C_i 中项目评分数目最多的前 m 个用户，再挑选出对项目集合 C_j 中项目评分数目最多的前 m 个用户，组成两个用户集合。如果前后两个用户集合中相同的用户越多，表示两个项目集合越相似；反之，则表示项目集合差异度越高。在算法运行过程中，将用户集 U 按照对项目集合内项目评分多寡由多到少重新排列，挑选出评分最多的前多少个项目构成用户集合，用于相似度的比较，由项目评分因子确定。

项目评分因子 K 是聚类算法的输入数据。当算法描述项目集合 C_i 的数据稀疏特征时，计算用户集 U 内各用户对项目集合 C_i 有评分的项目数目，按照用户对集合内有评分的项目个数由多到少排列，选择排在最前面的 K 个用户，即项目评分因子规定的数目，组成用户集合 $U_i=\{u_1,u_2,\cdots,u_k\}$。

在推荐系统的研究过程中，设计项目相似性计算公式和项目聚类算法时，常常考虑到项目类别的

影响。项目类别在一定程度上可以反映用户消费的偏好,如果用户对某类别的商品给以较高的评分,则其很有可能给同一类别的其他项目相应的评分。如果两个项目属于的项目类别越多,它们之间的相似性就应该越高。从项目聚类的角度,项目类别是一个天然的聚类依据,通过项目类别可以迅速将项目集划分为若干类。ICBSDIG 项目聚类算法在构建项目集合差异度时,将项目类别因素考虑其中。项目类别对项目集合差异度的影响是通过项目类别相似度 G 来实现的。在集合差异度计算公式中,项目类别相似度 G 出现在分母位置,项目类别相似度 G 越大,集合差异度越小。项目类别相似度 G 由项目集合中的项目所属项目类别的相似程度决定的。

与 K-means 等分割聚类算法不同,IBCRA 项目聚类算法通过项目的一次聚类,即可得到聚类结果。IBCRA 算法聚类次序按照各项目评分用户数目由多到少依次聚类,后文实验证明这种聚类顺序相对于项目集默认按照项目编号的排列顺序进行聚类,其效果更好。其他的一些基于聚类的协同过滤推荐算法研究文献也证明在项目聚类过程中,评分用户多的项目首先开始聚类有益于提高聚类质量。如王辉等[26]在使用 K-means 算法对评分数据集进行用户聚类时,以访问量(有效数据)最多的 k 个用户作为初始的 k 个聚类中心,经过实验验证可以较好地减少孤立点。Quan 等[27]设计的 Group By USim Stability 项目聚类算法,就是将项目集按照评分用户个数由多到少排列,选择评分用户数排在前 10%的项目首先开始聚类。

算法的输入数据、输出信息和主要步骤如下:

输入:用户—项目评分矩阵 R,项目类别矩阵 S_{item},项目类别因子 I,项目评分因子 K,集合差异度阈值 d,项目类别总数 G。

输出:项目聚类簇 $Cluster$。

1. 设项目集 N 中项目总数为 n,依次计算项目 I_1,I_2 直至 I_n 评分用户数目,并按照评分用户数目从大到小排列,项目重新排列后得到的项目集记作 N'。

2. 首先为项目集 N' 中的每一个项目创建一个初始项目集合 $C_i^{(0)}$,$i\in\{1,2,\cdots,n\}$。每个项目集合中只有一个项目,如项目集 $C_1^{(0)}$ 对应项目集 N' 中的第一个项目。

3. 计算项目集合 $C_1^{(0)}$ 与项目集合 $C_2^{(0)}$ 的集合差异度,计算公式为:

$$\mathrm{SFD}(C_1^{(0)},C_2^{(0)})=\frac{K-I}{K\times G} \tag{1}$$

如果 $\mathrm{SFD}(C_1^{(0)},C_2^{(0)})$小于集合差异度阈值 d,则将项目集合 $C_1^{(0)}$ 与 $C_2^{(0)}$ 合并,合并后的新项目集合记作 $C_1^{(1)}$;如果 $\mathrm{SFD}(C_1^{(0)},C_2^{(0)})$超过集合差异度阈值 d,则将 $C_1^{(0)}$ 与 $C_2^{(0)}$ 作为两个新的项目类,新的项目类记作 $C_1^{(1)}$ 与 $C_2^{(1)}$。项目类的个数记作 m。

4. 对于项目集合 $C_3^{(0)}$,计算 $\mathrm{SFD}(C_3^{(0)}\cup C_2^{(1)})$,$i=\{1,2,\cdots,m\}$寻找 i_0,使得

$$\mathrm{SFD}(C_3^{(0)}\cup C_{i_0}^{(1)})=\min_{i\in\{1,2,\cdots,m\}}\mathrm{SFD}(C_3^{(0)}\cup C_i^{(1)}) \tag{2}$$

如果 $\mathrm{SFD}(C_3^{(0)}\cup C_{i_0}^{(1)})$小于集合差异度阈值 d,则将 $C_3^{(0)}$ 与 $C_{i_0}^{(1)}$ 合并,新类仍然记作 $C_{i_0}^{(1)}$;反之,则将 $C_3^{(0)}$ 作为一个新类,即作 $C_{m+1}^{(1)}$;$m+1\Rightarrow m$。

5. 对于集合 $C_j^{(0)}$,$j\in\{4,5,\cdots,n\}$,依次重复步骤(4),进行相应的类别划分。

6. 通过项目集合的合并后得到的项目集合构成项目聚类簇 $Cluster$。项目聚类簇中包括 m 个项目类,每个项目类中记录了属于该项目类的项目编号。将孤立点的类从终类中去除,得到的即为最终聚类结果。

2.2 项目相似性计算

协同过滤推荐通过预测目标用户对未评分项目的评分,选择预测评分最高的项目推荐给用户;如

果是 $Top\text{-}N$ 推荐，则选择 N 个评分最高的项目列表推荐给用户[22,28]。计算预测评分的方法主要有以下两种[29]：(1)加权平均法，即通过计算用户对目标项目 i 评分相似的项目的评分加权和得到，权重为项目 i 和项目 j 之间的相似度 $\mathrm{sim}(i,j)$；(2)回归方法，该方法不是直接利用相似项目的评分值，而是基于回归模型得到的近似评分值。在实际运用中，因为两个欧式距离比较大的评分矢量之间很可能会有比较大的相似度，因此通过余弦法或者相关系数法得到的相似度可能不太准确，用邻居项目的原始分数来计算预测值的效果可能会比较差。回归方法采用用户基于回归模型得到的近似评分值而不是原始的评分值。回归模型可以表示为[29]：

$$\bar{R}_n = \alpha \bar{R}_i + \beta + \varepsilon \tag{3}$$

其中，$\bar{R}_i$ 和 $\bar{R}_n$ 分别是目标项目 i 以及邻居项目 n 的评分矢量，α 和 β 为回归模型参数，ε 是回归模型误差。

同回归方法相比，加权平均法的研究比较成熟，是当前绝大多数协同过滤推荐研究采用的评分预测方法。本文也采用加权平均法作为目标用户对未知项目的评分的预测方法。

加权平均法根据用户对目标项目邻居项目集合内的项目评分加权求和得到，权重由邻居项目同目标项目的相似度构成。用户 u 对项目 i 的预测评分 $r_{u,i}$ 的计算公式为[29]：

$$r_{u,i} = \frac{\sum_{j \in M_i} \mathrm{sim}(i,j) \times R_{u,j}}{\sum_{j \in M_i} |\ \mathrm{sim}(i,j)\ |} \tag{4}$$

其中，$R_{u,j}$ 表示用户 u 对项目 j 的评分，$\mathrm{sim}(i,j)$ 表示项目 i 和项目 j 的相似性，M_i 表示目标项目 i 的最近邻居项目集合。

在一些协同过滤推荐算法的研究文献中，在评分预测方法中考虑到用户评分尺度不同的影响，在基于项目的评分预测中，将项目评分均值添加到评分预测公式中。考虑项目评分均值的评分预测公式如下所示[11]：

$$r_{u,i} = \bar{R}_i + \frac{\sum_{j \in M_i} \mathrm{sim}(i,j) \times (R_{u,j} - \bar{R}_j)}{\sum_{j \in M_i} |\ \mathrm{sim}(i,j)\ |} \tag{5}$$

公式中，$\bar{R}_i$ 表示用户对项目 i 评分的均值，$\bar{R}_j$ 表示用户对项目 j 评分的均值。

式(4)是经典的评分预测公式，Sarwar[8]等在提出基于项目的协同过滤推荐算法的文献中，使用的评分预测公式即为式(4)。式(5)是在式(4)的基础上改进而来，但是目前许多研究文献仍然选用公式(4)，主要原因是考虑评分均值后的评分预测公式并没有表现出明显的优越性。因而，本文也选用公式(4)作为评分预测公式。

2.3 最近邻居查询

协同过滤推荐算法通过搜索与目标项目 j 相似度最高的若干个项目，构成目标项目的最近邻居项目集合 M_j。通过用户对目标邻居项目评分的加权平均值来描述用户对目标项目的评分，实现对未评分项目的预测评分。最近邻居查询是评分预测的基础，搜索形成的最近邻居集合的质量是决定预测评分准确性的最主要因素。

目标项目的最近邻居项目集合 M_j 的形成主要取决于两个因素：一是项目相似性计算方法。不同的项目相似性计算方法会影响项目与目标项目的相似性计算结果，进而影响最近邻居项目集合内项目排列顺序和权重。二是最近邻居查询个数。协同过滤推荐算法在最近邻居查询采用 K-nearest

思想,最近邻居查询个数是需要设定的参数,决定了最近邻居项目集合 M_j 的长度。本文用 *Knear* 参数来代表最近邻居查询个数。

最近邻居查询个数必须慎重设定,如果最近邻居查询个数赋值不当,将会影响协同过滤推荐结果。如果最近邻居查询个数过小,预测评分值受一两个评分数据的影响很大,容易因异常值影响到推荐效果;反之,如果最近邻居查询个数过大,除增加计算量外,还会因新增加的项目与目标项目相似性过小而影响推荐结果的准确性。

基于项目的协同过滤推荐算法将最近邻居的搜索范围限定在目标项目所属的项目类内,同在全项目集中搜索邻居项目,降低了计算量。项目聚类簇 *Cluster* 是项目聚类的结果,保存了划分的项目类和每个项目类内包含的项目。在对目标项目 j 进行评分预测时,首先从项目聚类簇 *Cluster* 中查找目标项目所属的项目类,并返回其序号 *index*。

在 *index* 类内,根据项目相似性计算得到的结果,按照邻居项目同目标项目 j 的相似性从高到低进行排列,得到的目标项目 j 的类内邻居项目集合 $M_j=\{j_1,j_2,\cdots,j_{Knear}\}$,$N_j$ 集合中的项目个数为 $\mathrm{Num}(N_j)$。

如果最近邻居查询个数 $Knear<\mathrm{Num}(N_j)$,选取类内邻居项目集合 N_j 中的前 *Knear* 个项目构成目标项目 j 的最近邻居项目集合 $M_j=\{j_1,j_2,\cdots,j_{Knear}\}$;如果 $Knear>\mathrm{Num}(N_j)$,表示目标项目的类内邻居项目个数尚不足 *Knear*,那么将集合 N_j 整体作为目标项目 j 的最近邻居项目集合,即 $M_j=N_j$。

如果目标项目 j 是孤立点,按照余弦相似性公式,需要在全项目集中计算目标项目同其他项目的相似性。同理,最近邻居查询也需要在全项目集中进行,并在全项目集 N 中查找出与目标项目相似性最高的 *Knear* 个项目,组成目标项目 j 的最近邻居项目集合 $M_j=\{j_1,j_2,\cdots,j_{Knear}\}$。

在实际工作中,还可能遇到一种情况:目标项目 j 不是孤立点,可以在目标项目所属的项目类内搜索到目标项目的 *Knear* 个或者 $\mathrm{Num}(N_j)$ 个邻居项目,构成最近邻居项目集合 M_j。但是,由于最近邻居项目集合中项目个数过少,在预测用户对未评分项目的评分的过程中,发现用户对目标项目的最近邻居项目集合 M_j 内的全部项目都没有评分。如果还按照项目类内进行协同过滤推荐,结果是用户对未评分项目的评分将为 0,这显然是不符合实际情况的。因而,当用户对目标项目的最近邻居项目集合 M_j 内的全部项目都没有评分时,必须更改目标项目的最近邻居查询规则,将目标项目等同于孤立点,改为在全项目集进行最近邻居查询。

2.4 评分预测过程算法

下面是在基于稀疏差异度和项目类别的项目聚类算法的聚类结果的基础上,通过在类内计算项目相似性和最近邻居查询,计算用户对未知项目的评分,产生最终推荐结果。

基于项目聚类的协同过滤推荐算法,最近邻居查询和评分预测算法主要步骤如下所示:

输入:用户—项目评分矩阵 R,项目聚类簇 *Cluster*,最近邻居查询个数 *Knear*

输出:目标用户 u 预测评分最高的项目 i 或者评分最高的 n 个项目(*Top-N* 推荐)

1. 针对目标项目 i,在项目聚类簇 *Cluster* 寻找目标项目 i 所属于的项目类 *index*。

2. 设 $I=\{i_1,i_2,\cdots,i_n\}$ 为推荐系统项目类 *index* 中全部项目的集合,则在集合 I 内计算项目 i 和项目 j($j\in I$ 且 $j\neq i$)的相似性,项目相似性计算使用余弦相似性公式。

$$\mathrm{sim}(i,j)=\cos(\vec{i},\vec{j})=\frac{\sum_{u\in U}R_{u,i}R_{u,j}}{\sqrt{\sum_{u\in U}R_{u,i}^2\sum_{u\in U}R_{u,j}^2}} \tag{6}$$

3. 如果目标项目 i 为孤立点，或者目标用户 u 对项目类 $index$ 类内其他项目都没有评分，则在全体项目集内计算项目 i 与其他项目的相似性。

4. 统计目标用户 u 在 $index$ 类中已经评分的项目的集合 I_u；如果 i 为孤立点，则集合 I_u 为目标用户 u 所有已经评分的项目的集合。

5. 计算目标用户 u 的未评分项目集，即没有被该用户评价的项目组成的集合 $I'_u=N-I_u$，其中 N 表示全体项目集合。

6. 根据计算得到的项目相似性计算结果和输入的项目最近邻居个数 $Knear$，计算目标项目 $i(i\in I'_u)$ 的 $Knear$ 个最近邻居，邻居项目组成集合 $M=\{i_1,i_2,\cdots,i_{Knear}\}$，相似度 $\{\mathrm{sim}(i,i_1),\mathrm{sim}(i,i_2),\cdots,\mathrm{sim}(i,i_{Knear})\}$ 从大到小的顺序排列。如果目标用户 u 已评分的项目总数小于参数 $Knear$，则集合 M_i 只选择 $\mathrm{Num}(Iu)$ 个最近邻居，$\mathrm{Num}(I_u)$ 表示项目类 $index$ 中项目总数。

7. 根据步骤(6)得到的目标用户 u 的最近邻居集合 M_i 和用户--项目评分矩阵 R 内的评分数据，依据用户 u 对目标项目 i 的最近邻居评分的加权平均值预测用户 u 对项目 $i(i\in I'_u)$ 的预测评分，即式(4)

$$r_{u,i}=\frac{\sum_{j\in M_i}\mathrm{sim}(i,j)\times R_{u,j}}{\sum_{j\in M_i}|\mathrm{sim}(i,j)|}$$

8. 重复步骤(6)和步骤(7)，预测目标用户 u 对所有的未评分项目的评分，选择预测评分最高的项目推荐给该用户；如果是 $Top\text{-}N$ 推荐，则选择评分最高的前 N 个项目推荐给用户。

3 实验验证及结果分析

3.1 数据集

本文选用的实验数据集来自 Minnesota 大学 GroupLens Research 项目组收集的 MovieLens 数据集(http://MovieLens.umn.edu/)，该数据集是当前绝大多数电子商务推荐系统研究选用的数据集。MovieLens 站点用于接收用户对电影的评分并提供相应的电影推荐列表，其评分尺度是从 1 到 5 的整数，数值越高，表明用户对该电影的偏爱程度高，反之则表明用户对该电影不感兴趣。

本文的实验数据集是在 MovieLens 数据集的全部 100 000 条数据中，筛去用户编号小于 500(用户编号的取值范围在 1 到 943)、项目编号小于 1 000(项目编号的取值范围在 1 到 1 682)，共得到 52 153 条评分数据。实验采用 5 折交叉验证法，将实验数据集平均分成 5 个互不相交的数据子集，训练集和测试集的数据比例为 4∶1。每次实验选择其中一个数据子集作为测试集，其余四个数据子集作为训练集。如此循环五次，取每次实验结果的平均值作为最终结果。5 折交叉验证法可以有效降低数据集对实验结果的影响。在 MovieLens 数据集中，评分数据是按照用户编号从低到高升序排列，同一用户的评分数据按照项目编号升序排列。在对实验数据进行划分时，为了保证不因实验者的偏好或数据集本身差异造成实验结果的失真，采用对评分数据打标记的方式。标记位的取值为 1 到 5 的整数，对于实验数据集的第一条评分数据标记为 1，第二条评分数据标记为 2，依次递增。当上一条评分数据的标记位已经取值为 5 时，下一条评分数据的标记位重新取值为 1，直到为实验数据集全部数据打标截止。根据打标后生成的标志位结果，对数据集按照不同的组合进行分类，得到实验训练集 1(train1)、实验测试集 1(test1)、实验训练集 2(train2)、实验测试集 2(test2)、实验训练集 3(train3)、实验测试集 3(test3)、实验训练集 4(train4)、实验测试集 4(test4)、实验训练集 5(train5)和实验测试

集5(test5),如表1所示.实验是在AMD Athlon (TM),64 Processor 3000+,1.8 GHz,1G Memory,Windows 2003 Server平台下,用Visual Studio C语言对算法编码进行实验的。

表1 数据集划分方法

数据集	标记位	数据条目	数据集	标记位	数据条目
train1	1 2 3 4	41722	test3	3	10430
test1	5	10431	train4	1 3 4 5	41723
train2	1 2 3 5	41722	test4	2	10430
test2	4	10431	train5	2 3 4 5	41722
train3	1 2 4 5	41723	test5	1	10431

MovieLens数据集中的项目是电影,根据电影的主题、性质等因素,将电影划分为19个类别,用0到18分别代表着19个项目类别(Item Category),依次为:

0: unknown	1: Action	2: Adventure	3: Animation
4: Children's	5: Comedy	6: Crime	7: Documentary
8: Drama	9: Fantasy	10: Film-Noir	11: Horror
12: Musical	13: Mystery	14: Romance	15: Sci-Fi
16: Thriller	17: War	18: Western	

MovieLens数据集的项目列表中列出了各个项目(每部电影)所属的项目类别。每个项目可能只属于一个项目类别,也可能同时属于多个项目类别,因为项目类别的划分标准是不一样的。根据项目所属项目类别信息,生成项目类别矩阵 $Sitem$。$Sitem$ 是 1000×19 维矩阵,两个维度分别代表项目和项目类别。如果项目 I_i 属于项目类别 g_j,则项目类别矩阵中,$Sitem_{ij}=1$;反之,如果项目 I_i 不属于项目类别 g_j,则 $Sitem_{ij}=0$。表2列出了 $Sitem$ 矩阵中的5行元素,分别代表项目1、项目2、项目3、项目4和项目5所属的项目类别。从项目类别矩阵中,可以清晰地表达出项目同项目类别间的对应关系,这是基于稀疏差异度和项目类别的项目聚类以及协同过滤推荐的重要输入信息。

表2 项目类别矩阵

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
I_1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
I_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
I_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
I_4	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
I_5	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0

3.2 推荐度量指标

评价推荐系统推荐效果的度量指标,主要包括统计精度方法和决策支持精度方法[28]。其中,统计精度方法——平均绝对偏差(Mean Absolute Error,MAE)作为评价推荐系统的推荐效果的度量标准被绝大多数推荐算法所采用。MAE方法通过度量推荐系统产生的对目标项目的预测评分与用户的实际评分之间的偏差反映推荐的准确性。算法工作在训练集上,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如公式(7)所示[28]:

$$\mathrm{MAE}=\frac{\sum_{i=1}^{N}|p_i-q_i|}{N} \tag{7}$$

一般情况下，MAE 指标的值越小，说明预测的评分和实际用户的评分相差很小，推荐质量越高。

3.3 试验结果

(1) 项目聚类算法

基于稀疏差异度和项目类别的项目聚类实验，聚类结果除受算法本身影响外，还取决于实验训练集数据，以及算法参数设置。在相同训练集数据情况下，项目类别因子 I、评分项目因子 K 和集合差异度阈值 d 的取值对聚类结果的影响很大。在实验开始前，项目类别因子、项目评分因子和集合差异度阈值需要作为常量设定。

算法工作在 train 1 上，设项目类别因子 $I=3$，评分项目因子 $K=25$，集合差异度阈值 $d=0.5$，得到的聚类结果如表 3 所示：

表 3 基于 IBCRA 算法的项目聚类实验结果

项目类	类内项目编号
0	50,286,181,294,174,300,121,117,222,172,405,313,173,210,276,204,69,25796,195,176,748,234,89,82,186,265,228,235,161,179,385,175,95,403,474,588,230,271,24,250,511,229,566,164,227,559,252,304,472,455,431,231,239,148,2,449,177,38,62,358,260,343,520,380,184,679,29,109,232,554,264,39,465,831,578,919,930,576,720,825,491,101,636,755,399,826,434,450,141,145,616,552,827,931,540,562,140,849,768,391,78,760,206,577,771,769,802,829,363,254,810,541,877,779,373,622,355,797,586,890,389,426,758,590,110,84,398,453,112,982,353,374,397,947,560,976,691,759,35,668 314,897,599,838
1	100,258,288,7,56,98,127,237,79,302,9,318,423,183,28,12,22,546,64,357,191,328,15,144,742,655,275,135,118,132,196,289,475,11,245,508,97,125,182,282,180,333,71,273,323,568,322,200,4,603,678,215,471,218,298,147,628,651,187,203,326,77,307,591,751,156,272,99,479,597,55,188,684,327,226,432,31,654,315,550,223,129,879,53,293,68,325,270,685,92,657,281,73,157,233,332,198,291,54,214,750,510,470,33,317,346,17,292,526,762,744,356,682,895,159,693,5,21,331,823,673,44,469,717,729,458,689,3,924,160,658,939,106,558,627,392,928,295,696,468,262,27,741,754,76,581,713855,686,934,128,943,595,365,619,329,466,61,743,772,887,977,366,583,881,840,841,649,809,886,896,975,244,761,833,880,339,370,898,43,620,344,876,961,963,974,572,350,984,571,752,460,979,336,467,937,330,925,308,972,832,46,808,459,986,348,457,349,985,279,375,703,149,359,916,980,454,933,337,6,587,915,146,983,555,888,37,726,74,534,899,883,927,912,803,798,891,981,994,536,909,807,917,920,776,867,784,817,851,711,852,296,987,830,626,910,868,911
2	1,151,269,168,202,25,216,111,238,268,70,194,185,153,435,197,143,483,496,211,8,433,301,514,209,340,134,154,208,14,515,124,88,284,527,23,393,133,427,274,411,845,42,199,367,480,58,86,402,137,732,285,13,523,83,66,283,319,462,248,476,150,321,419,451,255,692,240,410,259,582,443498,47,739,241,482,484,94,660,663,249,193,87,91,123,303,93,485,763,531,217,690,866,116,72,705,735,310,347,746,246,67,190,213,507,631,205428,65,528,504,478,509,382,699,178,521,136,642,170,126,659,242,747,792,421,212,192,162,815,290,708,381,425,430,81,316,629,59,715,152,51,429,529,709,378,10,90,517,652,221,781,518,604,306,131,52,506,710,549,736,45,163,414,461,502,544,280,724,727,155,158,49,650,737,486,499,305640,778,20,287,19,873,354,26,312,277,606,633,417,988,171,584,648,662,707,207,492,497,607,647,676,409,396,387,516,251,311,716,847,959,278,553,609,712,487,166,614,371,60,602,165,785,416,494,702,481,220,923,345,632,85,955,57,610,490,297,846921,952,956,524,236,794,495,856,694,697,789,872,941,948,949,107,512,503,875,664,683,765,522,740,783,796,641,224,612,16,464,842,535,942,945,704,723,738,749,764,639,882,990,731775,638,869,944,950,753,30,900,960,966,613,995,805,618,863,905,580,936,821,953,965,971,532,865,962,617,835,836,579,745,938,733,253,36,889,903,904,906,543,545,714,256,394,18,787,695,341,908,870,786,730,837,698,557,34,967,793,782,958970,918,822,113,598,643,964,594,957,907,913

续表

项目类	类内项目编号
3	596,501,418,243,408,473,225,169,404,926,756,338,412,122,477,401,386,625,871,420,969,63,780,820,105,189,384,120,864,108,114,946,80,369,687,722,142,395,102,790,538,167,892,575,456,993,40,688,721,372,542,824,139,383,364,342,843,407,951,929,940,728,500,368,989,585,41,415,539,388,819,261,932,734,624,335,812,623,376,422,719,878,818,894,138,795.997,998,390,862,400,718,103,801,377,700,548,996,901,725,791,893,828,104,600
4	324,525,513,493,334,770,488,505,615,489,656,608,611,902,653,978,574,592,914,130,533,848,361
5	48,32,530,463,813,519,644,634,645,115,320,811,360,766,701,954,884,85075,119,757,814,973,857,677
6	447,219,436,201,448,665,452,672,413,569,561,299,441,406,675,806,874,674,774,671,444,670,379,800,573,637,537,567,564,834,635,565,563,816,445,853,859,551,854,773,424,767,351,681,999,446,669,860,667,804,885,439,440,438,706,437,442,777,839,666,861,858,788,992
7	661,589,570,922,646,556,621
8	844,680,547,266,309,991,263,247
9	605,362,630,968,352
10	593,601,935,799
11	267

在训练集 train1 上,在上述指定参数下,得到的聚类结果分为 12 个类(类 0 到类 11),平均每个项目类包含 83 个项目。

在基于 IBCRA 项目聚类算法中,影响项目聚类簇结果主要是三个参数:项目类别因子 I、项目评分因子 K 和集合差异度阈值 d。其中,集合差异度阈值 d 对项目聚类结果的影响最为明显。在算法应用过程中,主要通过调节集合差异度阈值 d 影响项目聚类结果。

一般情况下,集合差异度阈值 d 的取值越高,不同的项目聚到一个项目类越困难,最终得到的项目类越多。d 值越接近于 0,在层次聚类过程中合并集合的尺度越松,项目间比较容易合并到一个项目类中。例如,在相同的参数条件下,同样工作在训练集 train1 上,当集合差异度阈值 d 取值为 0.4 时,得到的最终聚类结果包括 31 个项目类,平均每个项目类包括 28 个项目,另外还包括 136 个孤立点;而当 d 值为 0.5 时,只有 1 个孤立点(项目 267)。

从项目在各类别中的分布情况看,当集合差异度阈值 d 取值比较小时,项目聚集度比较高,生成的终类数目比较少。此外,编号靠前,即生成时间较早的项目类中项目个数普遍比较多;编号靠后的项目类中项目个数相对较少。这是因为当集合差异度阈值 d 取值较小,待聚类项目同各项目集合的最小集合差异度超出阈值 d 的可能性就比较小,按照 IBCRA 算法的聚类约定,待聚类项目将和集合差异度最小的项目集合合并,不会生成新的项目集合,最终形成的项目类数目比较少。

在 IBCRA 项目聚类过程中,项目聚类顺序也会对聚类结果产生明显的影响。在层次聚类算法中,聚类结果的质量受对象聚类顺序影响比较大,由于 IBCRA 算法属于层次聚类中的聚结型层次聚类,因而聚类次序同样对聚类质量产生重要影响,在算法设计中需要认真考虑。本文的 IBCRA 聚类算法在项目集合合并过程中,按照有评分的用户个数,从大到小排列依次聚类,是通过实验比较后选择的。第 4 节对比实验显示了项目聚类顺序对聚类质量的影响。

(2) 项目相似性计算

基于项目聚类的协同过滤推荐实验,是在基于 IBCRA 项目聚类算法的基础上,利用 IBCRA 项目聚类实验得到的项目聚类簇,在训练数据集 train$_i$,$i\in\{1,2,3,4,5\}$上通过项目相似性计算、最近邻居

查询和对未知项目的评分预测进行协同过滤推荐。项目相似性计算和最近邻居查询限定在同目标项目同属于一个项目类的项目。当目标项目属于孤立点或者目标用户对于与目标项目同属于一个项目类内的其他项目都没有评分的情况下，则在全体项目集内进行项目相似性计算和最近邻居查询。通过与对应的测试集 $test_i, i \in \{1,2,3,4,5\}$ 内各个数据，计算并比较预测评分与实际评分的差值，得到平均绝对误差 MAE 值。在完成 5 折交叉验证实验后，取 5 次实验得到的 MAE 值的均值$\overline{MAE}$，作为最终的实验结果。

基于项目聚类的协同过滤推荐实验，需要在基于 IBCRA 算法的项目聚类实验的基础上进行，实验输入除训练集、测试集数据以及需要指定的参数外，还需要 IBCRA 算法的项目聚类实验得到的项目聚类簇。实验具体的输入信息如下所示：

A 实验训练集数据 $train_i, i \in \{1,2,3,4,5\}$

B 实验测试集数据 $test_i, i \in \{1,2,3,4,5\}$

C 项目聚类簇 *Cluster*

D 最近邻居查询个数 *Knear*

基于项目聚类的协同过滤推荐实验得到的实验结果，是基于各训练集及对应测试集数据的 MAE 值，5 次测试得到的 MAE 值的均值$\overline{MAE}$作为衡量协同过滤推荐算法优劣的主要指标。$\overline{MAE}$值越小，协同过滤推荐算法的精度越高。

在基于项目聚类的协同过滤推荐实验中，设定项目类别因子 $I=6$，项目评分因子 $K=20$，集合差异度阈值 $d=0.7$，最近邻居查询个数 $Knear=20$。得到的实验结果如表 4 所示：

表 4 基于 IBCRA 项目聚类的协同过滤推荐实验结果

Knear	test1	test2	test3	test4	test5	$\overline{MAE}$
5	0.911 9	0.918 1	0.926 6	0.908 2	0.732 5	0.879 5
10	0.860 9	0.870 9	0.875 1	0.860 9	0.714 8	0.836 5
15	0.843 7	0.849 9	0.854 1	0.842 7	0.719 1	0.821 9
20	0.836 4	0.840 4	0.844 3	0.835 3	0.726 9	0.816 7
25	0.832 1	0.835 8	0.840 6	0.831 0	0.734 0	0.814 7
30	0.830 9	0.832 6	0.837 4	0.828 4	0.741 3	0.814 1
35	0.829 5	0.830 9	0.835 7	0.827 1	0.747 8	0.814 2
40	0.829 2	0.830 2	0.834 8	0.826 3	0.754 9	0.815 1
45	0.829 6	0.830 6	0.835 2	0.826 8	0.760 6	0.816 6

从上表中的实验结果中可以看到，基于 IBCRA 项目聚类的协同过滤推荐算法的推荐结果的平均绝对误差，除了受算法设计中的相似性计算、最近邻居查询规则、评分预测方法的影响外，还受到最近邻居查询个数 *Knear* 参数取值以及实验数据集的影响。其中，最近邻居查询法则是协同过滤推荐算法规则本身决定的，本文只是根据基于项目聚类的协同过滤推荐的一些具体问题，在孤立点、类内项目个数过少等情况下进行了相应的规则设定，没有对最近邻居查询规则做很大的改变。项目相似性计算和评分预测计算方法是影响推荐效果的重要因素，但不是基于项目聚类的协同过滤推荐的研究重点。基于项目聚类的协同过滤推荐算法的研究重点是设计科学的项目聚类算法，通过项目聚类降低数据稀疏性对项目推荐的影响，并以此提高推荐算法的精度。本文选择采用余弦相似性计算方法和加权平均评分预测法，是在此前研究文献对相似性和预测方法的介绍和对比的基础上做出的选择。

实验数据集作为外部因素也会对实验结果产生影响。从上表的实验结果中可以看到，算法工作在 train5 和 test5 得到的 MAE 值明显小于前四组实验集和测试集。一般而言，实验数据集对算法的

影响具有普遍性,即如果推荐算法 A 在某一训练集和测试集上得到的推荐结果的 MAE 值明显小于其他组,则推荐算法 B 在该训练集及测试集上得到的推荐结果的 MAE 值也明显小于其他组。例如,在 4.1 节采用 K-means 算法进行项目聚类,并在此基础上进行协同过滤推荐,在 test5 上计算得到的 MAE 值也明显小于其他四组。

最近邻居查询个数 *Knear* 对推荐结果和推荐算法的效率影响明显。4.2 节将详细阐述 *Knear* 值对推荐结果的影响。应用聚类算法,在项目聚类结果基础上进行协同过滤推荐,可以有效提高最近邻居查询的效率,但推荐质量受到聚类效果的明显影响。良好的聚类结果一方面可以降低协同过滤推荐计算的复杂度;另一方面可以整体提高推荐算法的推荐精度。实验结果上,在取不同的 *Knear* 值时,在不同的训练集和测试集上,预测结果的 MAE 值都有所下降。

4 对比试验

本节共介绍了三个对照实验。对照实验 1:基于 K-means 项目聚类的协同过滤推荐实验;对照实验 2:全项目集协同过滤推荐实验;对照实验 3:项目聚类顺序对推荐结果影响分析实验。对照实验 1 的目的,是为了比较本文提出的 IBCRA 项目聚类算法同当前国内本领域研究使用最多的项目聚类算法,即 K-means 算法的聚类效果进行比较,通过在基于各自算法得到的聚类结果上进行协同过滤推荐,比较预测评分的平均误差来评价 IBCRA 算法同 K-means 算法的优劣。对照实验 2 同样是为了验证 IBCRA 项目聚类算法的聚类结果,比较对象是不经过项目聚类,而在全项目集进行协同过滤推荐得到的推荐精度。对照实验 2 一方面可以验证 IBCRA 算法的聚类效果,同时也可以验证项目聚类对协同过滤推荐的积极作用。对照实验 3 的目的是验证在项目聚类过程中,项目聚类顺序对结果的影响,该对照实验选择了按照项目编号顺序聚类和按评分用户个数从多到少排列两种聚类顺序。

4.1 基于 K-means 项目聚类的协同过滤推荐实验

K-means 聚类算法是分割聚类的典型算法,在众多聚类应用中被广泛采用。K-means 算法由于其简单易行,在当前协同过滤推荐算法的研究中使用广泛,在诸多文献中被作为项目聚类、用户聚类、用户和项目混合聚类的聚类工具。K-means 算法的主要思想是:预先指定聚类的个数 k,将 n 个聚类对象分为 k 个类。首先随机选择 k 个对象代表 k 个类,每个对象作为该类的中心。而后,根据距离中心最近的原则将其他对象分配到各个类中。初次分配完成后,以每个类中所有对象的各属性均值作为该类新的中心,进行对象的重新分配。重复上述过程,直到聚类结果不再发生变化为止,即为最终的聚类结果。

K-means 聚类算法使用距离作为差异度的度量工具。传统的距离度量方法有欧式距离、绝对值距离、明考斯基距离等[10]。但是,从实验的结果分析,在使用 K-means 算法进行项目聚类,采用传统的距离度量方法效果比较差。在本文对项目编号在 1 000 以内的 999 个项目进行聚类时,无论是采用欧式距离,还是采用绝对值距离,都会出现大量项目聚集在一个项目类的情况,不能很好地实现项目聚类的目标。因而,基于 K-means 的协同过滤推荐算法的研究人员选择项目相似性计算公式度量项目间距离[10]。本文选择相关相似性度量项目间的相似性,并以相似性计算结果作为项目间距离的度量值。

在使用 K-means 算法对项目集进行聚类时,初始聚类数目 K 值需要事先指定,K 值的选取对实验结果有比较明显的影响。基于 K-means 项目聚类算法的协同过滤推荐实验的实验结果如表 5 所示。

表 5 基于 K-means 项目聚类的协同过滤推荐实验平均绝对误差值

Knear	*K*=5	*K*=10	*K*=15	*K*=20
5	0.878 2	0.911 0	0.960 9	0.966 7
10	0.844 0	0.895 6	0.925 8	0.957 3
15	0.840 8	0.890 5	0.934 6	0.960 2
20	0.839 2	0.895 2	0.932 2	0.968 3
25	0.840 1	0.896 8	0.951 4	0.963 0
30	0.845 9	0.898 2	0.939 9	0.958 1
35	0.850 5	0.895 9	0.951 4	0.967 2
40	0.851 7	0.900 8	0.948 4	0.970 3

将基于 K-means 聚类的协同过滤推荐结果，同基于 IBCRA 聚类的协同过滤推荐结果进行比较。其中 IBCRA 项目聚类设定项目类别因子等于 6，项目评分因子为 20，集合差异度阈值为 0.7。得到的结果如图 1 所示。

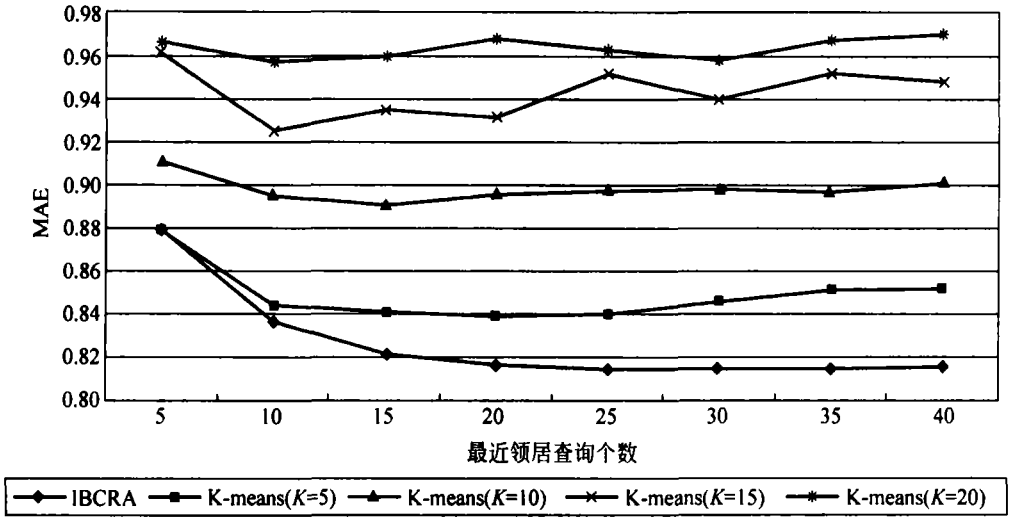

图 1 基于 K-means 与基于 IBCRA 项目聚类的协同过滤推荐结果对比图

从图 1 中可以清楚地看到，在使用 K-means 算法进行项目聚类并应用聚类结果进行协同过滤推荐时，实验结果随 K 值不同变动很大。当 K 值设定过大时，推荐精度不理想。同 IBCRA 算法得到的推荐结果项目，使用 K-means 算法进行的项目聚类由于 K 值不同得到的四组实验结果，即使是实验效果最好的一组（当 $K=5$ 时），推荐精度也要低于 IBCRA 算法。实验证明，基于稀疏差异度和项目类别的 IBCRA 算法的聚类结果和评分预测结果要优于传统的基于分割聚类的 K-means 算法。

此外，当最近邻居查询个数取值较大时，K-means 项目聚类结果曲线有明显的“上翘”现象，表明当最近邻居查询个数比较大时，随着最近邻居查询个数的增大，推荐精度反而有比较明显的下降。K-means 项目聚类的四组实验结果（$K=5$，$K=10$，$K=15$，$K=20$）推荐精度最高点，即平均绝对误差 MAE 值最小的实验点基本在最近查询邻居个数为 15 到 20 之间。当最近邻居查询个数超过 20 时，新增加的邻居项目同目标项目的相似性比较低，所以通过加权得到的预测评分值失真，推荐精度下降；但是，通过基于 IBCRA 项目聚类和未进行项目聚类的协同过滤推荐实验发现，当最近邻居查询个数超过 20 时，推荐精度仍有提高的空间。实验说明 K-means 算法没有能够充分地将相似度最高的项目聚为一类，造成目标项目所属类内同目标项目相似度高的项目个数比较少。

4.2 全项目集协同过滤推荐实验

4.1 节对照实验验证了 IBCRA 项目聚类算法同传统的 K-means 项目聚类算法相比，聚类效果更好。本节的全项目集协同过滤推荐实验的目的，就是为了比较基于项目聚类结果的协同过滤推荐算法和不进行项目聚类、在全项目集上进行协同过滤推荐二者的推荐效果。

全项目集协同过滤推荐实验在全体项目集而非与目标项目同属于一个项目类的项目间计算项目相似性，搜索最近邻居并进行评分预测。为了保证实验结果的对照效果，全项目集协同过滤推荐实验同基于项目聚类的协同过滤推荐实验一样，均采用余弦相似性计算项目相似性，同时采用 2.4 节中评分预测公式(4)预测用户对未评分项目的评分，并采用 3.2 节的平均绝对误差 MAE 公式(7)作为评价指标。实验采用 5 折交叉验证，训练集和测试集数据抽取和数据集划分方法和基于项目聚类的协同过滤推荐实验完全相同。

全项目集协同过滤推荐实验经过 5 折交叉验证后，得到的实验结果如图 2 所示。

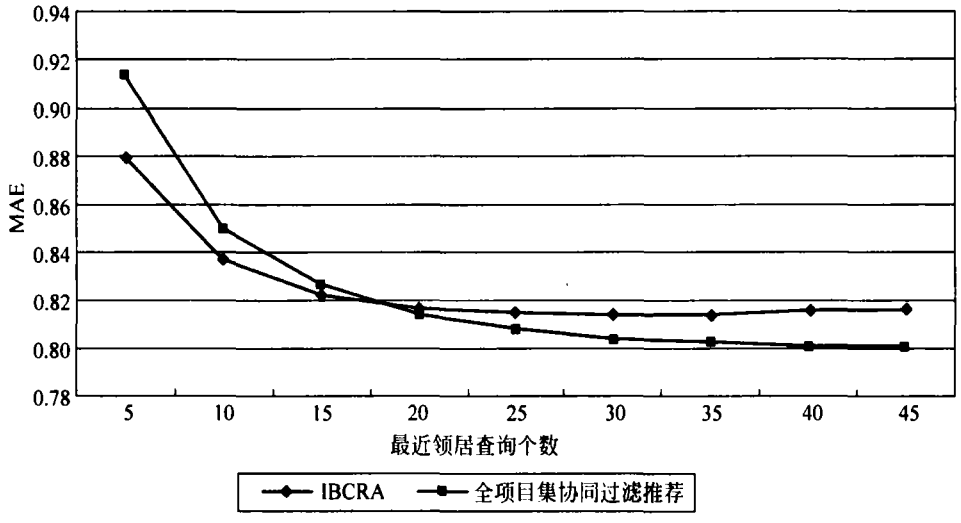

图 2 基于 IBCRA 项目聚类与全项目集协同过滤推荐实验结果对比图

从图 2 中可以看出，当最近邻居查询个数小于 20 时，基于 IBCRA 项目聚类的协同过滤推荐算法同在全项目集进行协同过滤推荐的算法相比，MAE 值较小，协同过滤推荐效果更好。从实验结果可以看出，在使用 IBCRA 算法对评分数据集进行项目聚类，可以有效地将相似度较高的项目划归一类，同一项目类内的项目间相似性较高。在类内进行项目间相似性计算，可以在一定程度上降低数据稀疏性对协同过滤推荐的影响，使得评分预测的结果更加有效。

当最近邻居查询个数超过 20 时，全项目集协同过滤推荐的精度开始超过基于 IBCRA 项目聚类的协同过滤推荐算法，但是彼此间差距比较小。当最近邻居查询个数为 45 时，基于 IBCRA 项目聚类的协同过滤推荐实验得到的 MAE 值超出全项目集协同过滤推荐实验最多，但也只超出了 2%。实验证明当最近邻居查询个数较少时，基于 IBCRA 项目聚类的协同过滤推荐算法推荐效果要优于未聚类结果；当最近邻居查询个数较大时，基于 IBCRA 项目聚类的协同过滤推荐算法的推荐精度基本可以达到未聚类时的水平。

在使用 IBCRA 算法对项目集进行聚类，并在聚类结果的基础上进行协同过滤推荐，使得项目相似性、最近邻居查询以及加权评分预测的计算范围由全项目集缩小到目标项目所属的项目类，使得项目搜索范围平均减少了超过 80%，大大提高了算法的效率，增大了研究成果实际应用的可能性。在推

荐精度上，当数据集、实验条件及主要参数设置完全相同时，基于 IBCRA 项目聚类的协同过滤推荐算法的推荐精度没有明显低于，甚至超过了传统的全项目集协同过滤推荐算法。因而，实验证明基于 IBCRA 项目聚类的协同过滤推荐算法能够兼顾算法的效率和效果，在一定程度上降低了先前基于聚类的协同过滤推荐算法追求算法效率对推荐效果的影响。

4.3 项目聚类顺序对推荐结果影响分析

IBCRA 项目聚类算法模仿了层次聚类算法的基本思想。在层次聚类算法中，影响聚类结果的一个非常重要的因素就是聚类的顺序。聚类顺序的不同选择会对聚类结果产生相当大的影响。为了验证聚类算法中聚类次序对聚类结果的影响，同时为 IBCRA 项目聚类算法选择较好的聚类输入次序提供实验依据，本文设计了项目次序对实验结果影响分析实验。

项目聚类顺序对结果影响分析实验设计了两种项目聚类顺序：一是按照项目编号的大小，从 1，2 直至 999 顺序聚类，这也是实验数据集初始的项目排列顺序。二是对项目集进行重新排序，排序的原则是按照项目的评分用户数目多寡，评分用户数目最多的项目排在最前，评分用户数目最少的项目排在最后，重新排列后的项目存储在 Ordering 一维数组。在使用 IBCRA 算法进行项目聚类时，聚类项目的输入顺序按照 Ordering 数组的项目排列顺序，从 Ordering[0]开始聚类，到 Ordering 数组最后一位存储的项目编号截止，聚类结束。为了保证实验结果的对照性，本实验除了在聚类过程中，项目输入次序不同外，聚类过程、项目相似性计算过程、最近邻居查询过程和评分预测过程完全相同，实验数据集和测试集的划分也完全相同。

5 折交叉验证的实验结果显示，本文在设计基于稀疏差异度和项目类别的项目聚类算法时，项目聚类顺序按照项目的评分用户数目排序，评分预测结果优于按照项目编号顺序聚类的评分预测结果。具体的实验结果数据如图 3 所示。

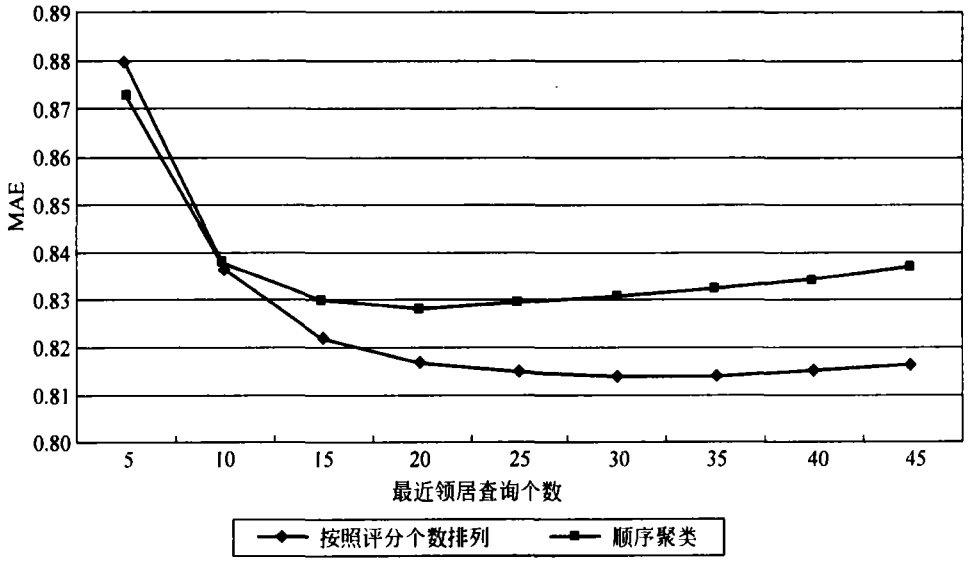

图 3 项目聚类顺序对推荐结果影响分析实验结果

在实验过程中，只有当最近邻居查询个数等于 5 时，顺序聚类得到的实验结果 MAE 值小于按照用户评分个数排列得到的 MAE 值，彼此相差 0.006 9。当最近邻居查询个数超过 5 时，顺序聚类得到的实验结果 MAE 值开始高于按照用户评分个数排列得到的 MAE 值，证明后者的推荐效果更好。而且，随着最近邻居查询个数的增加，按照评分用户个数排列和顺序聚类二者间的差距逐渐增大，从图 3

中明显可以看到两条曲线间的垂直距离越来越大。

项目聚类顺序影响聚类结果,直至影响协同过滤推荐结果的主要原因是IBCRA算法在聚类过程中定义集合差异度主要是依靠项目类别和评分数据的稀疏差异度。其中,评分数据的稀疏差异度构成了集合差异度计算公式的主体,而项目类别起到了调和的作用。算法评价集合稀疏差异度,主要是测算待聚类项目的评分用户同项目类内项目评分数目最多的前 K 个用户的吻合度(K 为项目评分因子)。由于各个项目间评分用户个数差距较大,部分项目有近200个用户评分,而有的项目只有极少的几个用户评分数据。按照项目编号顺序进行项目聚类,由于项目编号排列在前的项目可能评分数据很少,造成个别数据对结果的影响比较大,进而影响后面的聚类过程。按照用户评分个数重新排列后进行项目聚类,使得拥有最多用户评分数据的项目首先开始聚类,可以减少个别评分数据对聚类结果的影响。

从实验数据集的角度看,调整项目聚类顺序可以被认为是通过调整项目维度的顺序,将密集的项目列前移,使得用户—项目评分矩阵的前部分变得密集,有效降低数据稀疏性对聚类结果的影响。聚类顺序排在最后的项目评分数据很稀疏,易对聚类结果的准确性造成影响。但是根据顺序聚类法的特征,聚类顺序偏后的聚类对象较聚类顺序靠前的聚类对象,对聚类结果的影响相对较小。因而将最容易造成聚类结果失真的项目放在最后聚类,可以提高聚类结果以及协同过滤推荐结果的准确性。

5 结论

实验验证部分设计了基于评分数据稀疏差异度和项目类别的聚类算法(IBCRA算法)的项目聚类实验和基于IBCRA项目聚类的协同过滤推荐实验两个主要实验,实现了实验数据集划分、数据集输入和转换、IBCRA项目聚类、项目相似性计算、最近邻居查询、评分预测、预测结果误差测定等基于项目聚类的协同过滤推荐算法的各个主要环节。通过5折交叉验证,得到了良好的实验结果。

通过与使用广泛的基于K-means项目聚类的协同过滤推荐算法比较,IBCRA算法在推荐精度上要优于K-means算法,聚类时间小于K-means算法。基于IBCRA项目聚类的协同过滤推荐得到的平均绝对误差在最近邻居查询个数小时,优于未经项目聚类而在全项目集上进行协同过滤推荐得到的实验结果;当最近邻居查询个数比较大时,二者差距不是很大。在使用IBCRA算法进行项目聚类时,项目聚类顺序对实验结果有比较大的影响。通过实验分析,本文采用的按照评分用户数目由多到少顺序聚类的实验效果优于按照项目编号顺序聚类的实验结果。

除项目聚类顺序外,算法主要参数设置对推荐效果也会产生明显的影响。其中,集合差异度阈值、项目类别因子、项目评分因子和最近邻居查询个数是主要的影响因素。通过上述四个参数的取值变化,可以调节项目聚类结果,并影响协同过滤推荐实验的平均绝对误差(本文没有详细阐述这部分内容,需要的读者可向作者索取)。

当前,个性化电子商务推荐系统的理论研究和实际应用仍然面对很多困难。本文对项目聚类和协同过滤推荐算法的研究虽然取得了一定的成果,但依然在算法的效率和推荐质量等方面存在着诸多需要改进的地方。未来,基于高维稀疏数据聚类的协同过滤推荐算法需要在以下几个主要方面做进一步的深入研究:

(1) 进一步降低推荐算法计算的复杂度。虽然同需要反复迭代的K-means算法相比,IBCRA项目聚类算法一次聚类便可以得到最终结果。但是,在计算用户评分项目个数、对用户集合重新排序等若干个环节,当用户—项目评分数据集迅速增大时,计算量随之增长,在一定程度上影响推荐算法的可扩展性。

(2) 算法的推荐精度仍有进一步提高空间。实验结果虽然表明 IBCRA 项目聚类结果好于 K-means 算法,但是在大数据量、数据集极度稀疏的情况下,基于 IBCRA 项目聚类的协同过滤推荐效果还有待进一步研究。

(3) 算法各参数对推荐结果的影响原因及规律分析有待进一步深化。本文在实验验证过程中,对集合差异度阈值、最近邻居查询个数、项目类别因子和项目评分因子进行了参数敏感性分析,得到了初步的结论。但是,由于推荐结果受多个参数综合影响,具体每一个参数对推荐结果的影响规律,还没有研究得非常透彻,特别是项目类别因子和项目评分因子的影响规律。此外,在参数敏感性分析过程中,还需要考虑到实验数据集对结果可能产生的影响。

参考文献

[1] 余力,刘鲁. 电子商务个性化推荐研究[J]. 计算机集成制造系统, 2004, 10(10): 1306-1313.

[2] Resnick P, Varian H R. Recommender systems[J]. Communications of the ACM, 1997, 40(3): 56-58.

[3] Schafer J, Konstan J, Riedl J. Recommender systems in e-commerce[C]. Proceedings of the 1st ACM conference on Electronic Commerce, Denver, CO, USA: ACM, 1999: 158-166.

[4] 刘鲁, 任晓丽. 推荐系统研究进展及展望[J]. 信息系统学报, 2008, 2(1): 82- 90.

[5] Schafer JB, Konstan J, Riedl J. Recommender systems in e-commerce [C]. Proceedings of the 1st ACM Conference on Electronic Commerce, Denver, Colorado, USA, 1999: 158 - 166.

[6] 黄巧莉, 刘胜, 刘飞. 网络化销售和定制个性化信息推荐系统研究及应用[J]. 现代制造工程, 2005, 8: 31-35.

[7] 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.

[8] Sarwar B, Karypis G, Konstan J, Riedl J. Analysis of recommendation algorithms for e-commerce [C]. Proceedings of 2nd ACM conference on Electronic Commerce, Minneapolis, Minnesota, USA: ACM, 2000: 158-167.

[9] 姚忠, 吴跃, 常娜. 集成项目类别与语境信息的协同过滤推荐算法[J]. 计算机集成制造系统 2008, 14(7): 1449-1456.

[10] 武森, 高学东, M 巴斯蒂安. 高维稀疏聚类知识发现[M]. 北京: 冶金工业出版社, 2003.

[11] Adomavicius G, Tuzhili A. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[12] 何光辉, 魏曙光, 王蔚韬. 改进的聚类邻居协同过滤推荐算法[J]. 计算机科学, 2004, 31(11): 147-149.

[13] Schein A, Popescul A, Ungar L, Pennock D. Methods and metric for cold-start recommendations [C]. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland: ACM, 2002: 253- 260.

[14] 欧立奇, 陈莉, 马煜. 协同过滤算法中新项目推荐方法的研究[J]. 微计算机信息, 2005, 21(11-3): 186-188.

[15] 彭玉, 程小平, 徐艺萍. 一种改进的 Item-based 协同过滤推荐算法[J]. 西南大学学报(自然科学版), 2007, 29(5): 146-149.

[16] 邢春晓, 高凤荣, 站思南等. 适应用户变化的协同过滤推荐算法[J]. 计算机研究与发展, 2007, 44(2): 296-301.

[17] Adomavicius G, Sankaranarayana R, Sen S. Incorporating contextual information in recommender systems using a multidimensional approach [J]. ACM Transactions on Information Systems, 2005, 23(1): 104-145.

[18] 张锋, 常会友. 使用 BP 神经网络缓解协同过滤推荐算法的稀疏性问题[J]. 计算机研究与发展, 2006, 43(4): 667-672.

[19] Huang Z, Chen H, Zeng D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering [J]. ACM Transactions on Information Systems, 2004, 22(1): 116-142.

[20] Aggarwal CC, Yu PS. Redefining clustering for high-dimensional applications [J]. IEEE Transactions on

Knowledge and Data Engineering, 2002, 14(2): 210-225.
[21] 孙多. 基于兴趣度的聚类协同过滤推荐系统的设计[J]. 安徽大学学报(自然科学版), 2007, 31(5): 19-22.
[22] Deshpande M, Karypis G. Item-based top-N recommendation algorithms[J]. ACM Transactions on Information Systems, 2004, 22(1):143-177.
[23] O'Connor M, Herlocker J. Clustering items for collaborative filtering[C]. Proceedings of the ACM SIGIR Workshop on Recommender Systems, New Orleans, Lousiana: ACM, 1999.
[24] 张海燕, 丁峰, 姜丽红. 基于模糊聚类的协同过滤推荐方法[J]. 计算机仿真, 2005, 22(8): 144-148.
[25] 高凤荣, 杜小勇, 王珊. 一种基于稀疏矩阵划分的个性化推荐算法[J]. 微电子学与计算机, 2004, 21(2): 58- 62.
[26] 王辉, 高利军, 王听忠. 个性化服务中基于用户聚类的协同过滤推荐[J]. 计算机应用, 2007; 27(5): 1225-1227.
[27] Quan TK, Fuyuki I, Shinichi H. Improving accuracy of recommender system by clustering items based on stability of user similarity[C]. International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), Sydney, Australia: IEEE Press, 2006: 61-68.
[28] Karypis G. Evaluation of item-based top-N recommendation algorithms[C]. Proceedings of 10th international conference on Information and knowledge management, Atlanta, Georgia, USA: ACM, 2001: 247-254.
[29] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms[C]. ACM WWW10, Hong Kong: ACM, 2001: 285- 295.

Collaborative Filtering Recommendation Algorithm under High Sparse Data Clustering

YAO Zhong, WEI Jia & WU Yue
(Department of Information Systems and Information Management, School of Economics and Management, BeiHang University, Beijing 100083)

Abstract In order to resolve the poor-quality of recommendation in collaborative filtering recommendation algorithms in case of the high sparse dataset, this paper proposes a novel algorithm named item-based clustering recommendation algorithm (IBCRA). One of characteristics in the IBCRA is that it has considered the properties of data sparse difference and item category clustering within user-item dataset. Specifically, on the basis of high-dimensions data clustering algorithms, the IBCRA algorithm uses the rating data sparse difference and item categories in the rating dataset to construct a measuring formula for calculating dataset difference, where the formula is used for item clustering in user-item rating array. Then the IBCRA calculates item similarity and searches for k-nearest neighbors of target item based on the outcome of item clustering. Finally it forecasts the ratings for those no rating item in dataset and so generates recommendations. The experimental results show the IBCRA has improved the recommendation quality in collaborative filtering recommendation. The comparative experiments and parameter sensitivity analysis also show, in perspective of the accuracy and speed of convergence, the IBCRA also outperforms the collaborative filtering recommendation algorithm with all items based algorithm.

Key words Recommender systems, Collaborative filtering, Item clustering, Item category ranking, IBCRA