分类号_____ 密 级_____

U D C _____

学 位 论 文

交互式协同过滤推荐方法的研究

作 者 姓 名 ： 王向阳

指 导 教 师 ： 张斌 教授

东北大学计算机科学与工程学院

申请学位级别： 硕士 学 科 类 别 ： 工学学位

学科专业名称： 计算机应用技术

论文提交日期： 2016 年 12 月 论文答辩日期： 2016 年 12 月

学位授予日期： 2017 年 1 月 答辩委员会席： 赵海

评 阅 人 ： 高岩、刘峰

东 北 大 学

2016 年 12 月

A Thesis in Computer Application Technology

Research on Interactive Collaborative Filtering Recommendation Method

By Wang Xiangyang

Supervisor: Professor Zhang Bin

Northeastern University

December 2016

独创性声明

本人声明，所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其它人已经发表或撰写过的研究成果，也不包括本人为获得其它学位而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

日　　期：

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流。

作者和导师同意网上交流的时间为作者获得学位后：

半年 □　　一年□　　一年半□　　两年□

学位论文作者签名：　　导师签名：

签字日期：　　签字日期

摘 要

日新月异的互联网与信息技术给用户带来了更多的信息和服务，产生了海量的数据，其中包含的大量垃圾信息也给用户带来了困扰，这就是“信息过载”问题。推荐系统能够帮助用户更好地匹配个人感兴趣信息，在解决“信息过载”问题上卓有成效。目前，协同过滤算法是推荐系统中应用最广泛、最成功的技术，它根据用户历史信息，找出用户间的相似关系进行推荐。但是，随着用户数量的不断增多，系统规模的逐渐扩大，传统协同过滤算法中的数据稀疏性问题、用户推荐反馈问题、基于用户反馈的推荐优化问题日益凸显，影响了推荐质量。针对上述问题，本文提出交互式协同过滤推荐方法。

（1）针对数据稀疏性问题，本文通过考虑用户自身的评分偏好以及用户间的评分差异，提出一种基于用户评分差异度信息熵的相似性计算（URDE）方法。该方法通过消除用户自身的评分偏好以及用户间的评分差异，引入信息熵来计算用户间评分的相似性，把用户共同评分项目的数量作为调节用户间相似性的权重。在MovieLens电影数据集上，将URDE方法与其它相似性计算方法进行对比实验，结果表明在数据稀疏性不变的情况下，本文的URDE方法能够提高推荐精度。

（2）针对用户推荐反馈问题，本文通过考虑用户不同的交互行为隐藏的兴趣信息不同，提出一种基于推荐结果的用户交互行为兴趣度计算方法。该方法根据用户交互行为的不同，将其分为基于项目的行为和基于类标签的行为，通过计算各自的用户兴趣度来衡量用户对项目以及用户对类标签的感兴趣程度。其中，基于项目的行为指用户与推荐列表之间的交互行为，通过使用用户最小交互行为组合兴趣度计算方法得到用户对项目的兴趣度矩阵；基于类标签的行为指用户与分类标签之间的交互行为，通过欧氏距离以及兴趣度计算公式得出用户对类标签的兴趣度矩阵。在此基础上通过用户兴趣度调整函数调整用户兴趣度矩阵，从而把用户推荐反馈问题转换成用户兴趣度的计算问题。

（3）针对基于用户反馈的推荐优化问题，本文通过所定义的用户兴趣度矩阵，提出一种基于用户交互行为兴趣度的相似性计算（UIBD）方法。该方法根据用户兴趣度矩阵的不同，分为基于用户项目兴趣度的相似性计算方法和基于用户类标签兴趣度的相似性计算方法，来衡量用户之间的相似性，最后将其进行组合，通过权重值的调整得到用户间相似性，从而把兴趣度应用到协同过滤的相似性计算中。将UIBD方法与其它相似性计算方法进行对比实验。结果表明本文的UIBD方法在推荐结果的基础上能够提高推荐精度。

关键词：协同过滤；交互式；差异度信息熵；相似性；兴趣度

Abstract

Network and information technology advance rapidly, which brings users more information and services as well as create massive amounts of data. However, users are troubled by a great quantity of spam contained in mass data. This is "information overload". The advent of recommendation system can help users easily get information that match with individual interests. It has made great achievements in solving "information overload" problems. Collaborative filtering algorithm is the most widely used and most successful technology in the recommendation system at present. According to history information, this algorithm finds the similarity relation among the users and provides recommendation to users. However, with number of users increasing, many problems arise and affect the recommendation quality, for instance, the scale of the system expansion, the problems of data sparsity, user recommendation feedback issue and recommendation optimization based on user feedback issue. Interactive collaborative filtering recommendation method is proposed to deal with these issues in this thesis. The main contributions of this thesis are as follows:

(1) To solve the problem of data sparsity, this thesis, with consideration of user's preference and the difference between the users, proposes a similarity calculation method which is based on the information entropy of the user's degree of difference (URDE). This method eliminates the difference of user's preferences and different ratings among users, based on this, the information entropy is introduced to calculate the similarity between users' ratings, and the number of users' common rating items is used as the weights to adjust the similarity between users.With experiment comparison of MovieLens movie data set and other similarity calculation method, this thesis points out that in the case of the same data sparsity, the URDE method can improve the recommendation accuracy.

(2) To solve the problem of recommendation feedback, a method to calculate the interest degree of user interaction behavior is developed in this thesis by considering the different interest information hidden in different user interaction behaviors. It is based on the recommendation result. This method can be divided into behavior based on item and behavior based on category labels according to different user’s interaction behavior. And it can measure user’s interest of the item and the label by calculating the respective user interest degree. The behavior based on item refers to the interaction between the user and the recommendation list,

the user interest degree matrix is formed by using the user minimum interaction behavior combination interest degree calculation method. Behavior based on category labels refers to the interaction between user and category label, the user interest degree matrix is formed by calculating Euclidean distance and the degree of interest calculation formula. And the user interest degree matrix is adjusted by the user interest degree adjustment function， the user recommendation feedback problem is converted into the calculation of user interest degree.

(3) To solve the problem of recommendation optimization based on user feedback, this thesis, by analyzing user interest matrix definition, put forward a similarity calculation method, on the basis of user interaction behavior interest degree (UIBD). This method is divided into two similarity calculation based on the differences of user interest matrix: Similarity calculation method based on user's item interest degree and the similarity calculation method based on user's category label interest degree. Both calculation methods are to measure the similarity between users. With the combination of these similarity, similarity between users can be measured by adjusting weights. Compared with other similarity calculation methods， the experimental result indicated that the UIBD method can improve the recommendation accuracy based on the recommendation results.

Key words: collaborative filtering; interactive; similarity; interest degree

目 录

第 1 章 引 言

本章主要对论文的研究背景与意义、国内外研究现状进行了介绍，指出目前协同过滤算法在推荐领域的重要性，引出本文的研究目的、主要研究内容并且对本文的组织架构进行了描述。

1.1 研究背景与意义

Internet的出现彻底改变了人们的生活方式，从整个历史的发展进程来看，其几十年的时间很短，但是却带来了深远影响。以信息检索的角度来看，Internet的发展大概分为以下三个过程。

（1）门户网站

在Internet刚刚出现的时候，网络资源相对匮乏，人们对网络的依赖程度并不是特别高，这个过程的信息检索方式往往是通过门户网站聚集的，并不能很好地满足人们的需求，例如国外的雅虎，国内的中国黄页。

（2）搜索引擎

随着人们对网络的认知越来越清晰，网络给人们带来了便利，使得这段时间积累了大量的用户，此时，Internet发展的速度超出人们的预期，通过门户网站对信息聚集的方式已经远远满足不了人们对信息检索的要求，产生了海量的数据，其中包含的大量垃圾信息给用户带来了困扰，这就是“信息过载”[1]问题，此时以搜索引擎为代表的信息检索技术[2]应运而生，能够对这一问题进行解决。搜索引擎大大减轻了人们检索其目标资源的工作量，它根据用户输入的关键字进行相关资源的搜索，返回给用户对应的相应网页。例如国外的谷歌，以及国内后来发展的百度。虽然搜索引擎在一定程度上能够缓解“信息过载”问题，但是搜索引擎对用户的要求较高，要求用户能够相对比较准确的通过关键字描述自己的需求，从而进行相关资源的检索，但矛盾的是用户有时对自己的需求并不明确，这种情况下的检索结果可能并不好。

（3）推荐系统

以推荐系统为代表的信息过滤技术[3]就是在这种环境下出现的，它能够更好地结合用户历史浏览行为、交易记录，从而更好的匹配用户个人感兴趣信息，在解决“信息过载”问题上取得了很大成效。信息过滤技术[4]解决此问题的主要方法是通过分析用户历史行为数据，发现用户隐藏的需求，对用户进行相关资源的推荐，在一定程度上提高了用

户体验。

1.2 国内外研究现状

协同过滤推荐算法通过分析用户的历史行为[4]，发现用户的隐藏的兴趣，进行相关服务的推送，为用户提供更好的服务。这些用户的行为包括用户的浏览记录、交易记录、评论记录等等。其中每个用户行为隐藏的用户兴趣度是不同的。协同过滤通过用户行为用户的历史行为包括：浏览，购买，搜索，评论，收藏，分享等。不同的行为表示了用户对于不同项目的喜好程度。协同过滤推荐算法是目前业界应用最广泛的推荐算法[5]。

协同过滤这一推荐技术最早是在上世纪末由Goldberg等学者专家提出的[6]。Goldberg等通过TaPestry系统对用户有用的电子信息进行过滤，此系统要求用户参与度比较高，需要用户标记对其作用比较大的信息。1996年，美国明尼苏达大学研究学者把协同过滤技术运用到实际系统当中，通过协同过滤技术搭建了一个电影推荐系统，叫做MovieLens，然后将它发布到网上，这是一个开创性的工作，目前，MovieLens仍然起到很大的作用，其系统公开的MovieLens数据集也为协同过滤算法的研究产生了深远影响。之后，协同过滤算法越来越普遍地被运用到各个领域中，并且起到了很好的作用，为人们的带来了更多便利。国外相对较知名的推荐系统有Jester笑话推荐系统；Video Recommendation CD、DCD推荐系统；Grouplens；网上新闻过滤系统；知名的亚马逊网上书店的图书推荐系统也应用了协同过滤算法。

进入20世纪以来，国外协同过滤技术的应用发展受到国内相关学者的重视，也成为推荐领域中的研究热点之一。国内研究者在涉及到推荐系统的各方面，包括智能数据挖掘[7]、统计数据分析[8]、资源数据处理[9]的精确度都取得了丰硕的成果。同时推荐系统已经在很多领域得到应用，其中比较著名的有豆瓣推荐、淘宝网[11]商品推荐等。

豆瓣网是比较有名的推荐系统。豆瓣网当中的“豆瓣猜”就是一个典型的推荐模块，如图1.1所示，它的推荐理念是应用口碑式的营销方法，了解用户的鉴赏喜好，即一个用户做最有效的选择必然会受到同事或者朋友的影响，豆瓣网的推荐精确度很高。同样应用广泛的淘宝网也使用其推荐技术为用户体验更好地服务。淘宝网也在不断优化自己的推荐技术，比如每年的“阿里天池大数据竞赛”，需要参赛者根据其提供的大量用户历史交易记录，去预测用户下一步的交易行为，进而进行推荐。淘宝网目前的推荐系统智能化程度很高，用户购买、浏览了一些商品后，会进行相关物品的推荐，并且成功率很高。一些著名的电子商务系统也采用了这样的推荐模块为用户提供个性化推荐，比如淘宝网和京东电子商城。

图 1.1 典型的推荐系统—豆瓣猜

Fig. 1.1 Typical recommendation system-broad bean guessing

总体来讲，国内主要是在理论层次对于协同过滤技术进行研究。一方面，研究起步较晚，另外，协同过滤自身有数据稀疏性[12]、冷启动[13]和实时性[14]等问题，所以，不仅要在理论上研究协同过滤技术，而且要加大实际生活中的应用。

1.3 研究目的

协同过滤推荐算法中也存在一些问题，例如稀疏性问题[15]、用户推荐反馈问题[16]、基于用户反馈的推荐优化问题，本文主要针对这些问题进行研究。

（1）稀疏性问题：当用户和项目的数量达到一定的程度，整体用户的行为记录又比较少时就会导致用户对项目的评分矩阵极度稀疏，从而大大降低用户对项目预测评分的准确性，这就是协同过滤推荐算法的稀疏性问题。目前解决这一问题的主要方法就是评分矩阵的数据填充[17]。由于这种方法是人为的进行数据的填充，人的主要性比较大，并不能代表原始用户对项目的评分喜好程度，存在很大的误差，是最终的推荐效果并不理想。目前的推荐依赖于大量的用户评分矩阵，但是当用户评价有限的时候，难以利用评分矩阵进行推荐。针对上述问题，本文通过考虑用户评分的特点。对基于用户评分差异度信息熵的相似性计算方法进行研究。

（2）用户推荐反馈问题：目前的推荐大都是一种静态的推荐，即根据用户的评分矩阵进行计算以推荐给用户新的可能感兴趣的项目。但是，推荐是一个交互式的动态过程。用户对推荐列表中物品的兴趣度将会反映用户兴趣度的变化，即用户对推荐结果的反馈，应该作为推荐列表更新优化的依据。为此如何解决用户的推荐反馈是一个关键问题。针对此问题本文对基于推荐结果的用户交互行为兴趣度计算方法进行研究。

（3）基于用户反馈的推荐优化问题：在用户推荐反馈得到解决之后，如何将反馈应用到整个推荐的过程中从而优化推荐成为一个关键问题，针对这个问题，本文对基于

用户交互行为兴趣度的相似性计算方法进行研究。

1.4 主要研究内容

本文针对于协同过滤的据稀疏性问题、用户推荐反馈问题、基于用户反馈的推荐优化问题，研究一种交互式协同过滤推荐方法。构建交互式协同过滤推荐框架，并给出具体的实现过程。该方法基于所定义的用户评分与交互行为兴趣相似度模型，对目前的相似性计算进行了扩展。针对用户间评分的差异，以及用户本身评分的偏好差异，研究基于用户评分差异度信息熵的相似性计算方法来衡量用户评分的相似程度，该方法主要是用户的评分矩阵进行差异化的处理，消除用户自身的评分偏好以及用户间的评分差异，通过引入信息熵的方法来进行用户间相似性的计算法，从而在一定程度上解决了数据稀疏性对相似性计算的影响，作用到整个的推荐过程中，从而产生更好地推荐结果。针对用户推荐反馈的问题，通过考虑用户不同交互行为包含用户兴趣信息不同，引入了兴趣度的概念，并在此基础上分别计算了用户项目兴趣度以及用户类标签兴趣度，形成用户兴趣度矩阵，从而把用户推荐反馈问题转化为用户兴趣度的问题。针对基于用户反馈的推荐优化问题，本文针对用户不同的交互行为研究基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，在此基础上，通过调整基于用户交互行为兴趣度的相似性计算方法中的权重，得到用户相似性，作用到协同过滤的推荐过程中，优化推荐。具体研究内容如下：

（1）交互式协同过滤框架的构建

本文在传统协同过滤的基础上对交互式协同过滤推荐方法进行研究，构建交互式协同过滤推荐框架，并给出具体的实现过程。该方法主要包括三个主要部分：数据预处理、用户相似性的计算、用户兴趣度的计算。数据预处理对原始的用户评分数据进行提取形成用户-项目评分矩阵；用户相似性计算主要缓解传统协同过滤算法中的数据稀疏性问题，以及给予反馈的推荐优化问题；兴趣度的计算主要是解决用户推荐反馈问题。

（2）基于用户评分差异度信息熵的相似性计算方法的研究

相似性计算在协同过滤推荐算法中一直是一个关键问题，但是目前算法中往往面临着评分数据稀疏的问题，这样用户之间的相似性往往会比实际的结果偏大或者偏小。针对数据稀疏性问题，本文通过考虑用户自身评价偏好、用户间评分差异对基于用户评分差异度信息熵的相似性计算方法进行研究。该方法针对用户自身的评分偏好，用户之间评分尺度的不同，习惯的不同，进行差异化的处理，通过信息熵的方法衡量用户评分的相似性，进而进行推荐。

（3）提出基于推荐结果的用户交互行为兴趣度计算方法的研究

针对用户的不同交互行为本文研究基于推荐结果的用户交互行为兴趣度计算方法。该方法将用户的行为分为基于项目的行为以及基于类标签的行为：基于项目的行为主要是指用户与推荐列表之间的交互行为，通过对用户交互行为兴趣度的计算形成用户兴趣度矩阵，通过用户最小交互行为组合兴趣度计算方法计算用户项目兴趣度矩阵用户项目相似性的计算；基于类标签的行为主要是指用户与分类标签之间的交互行为，通过欧氏距离计算得出用户类标签兴趣度矩阵，用于用户类标签相似性的计算。该方法将用户的交互行为转化为用户的兴趣度矩阵，并应用到协同过滤的推荐过程中，解决用户推荐反馈问题。

（4）基于用户交互行为兴趣度的相似性计算方法的研究

针对用户项目相似性与用户类标签相似性本文研究基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，分别针对用户项目兴趣度矩阵和用户类标签兴趣度矩阵进行计算得到对应的用户相似性，该方法分别针对用户项目兴趣度矩阵和用户类标签兴趣度矩阵进行相似性的计算，从而优化相似性计算的方法，解决基于用户反馈的推荐优化问题。

1.5 论文组织结构

本文分为七个章节，具体章节结构如下：

第1章引言。主要对推荐技术的背景与意义、国内外研究现状进行了介绍，在目前协同过滤存在的问题的基础上指出了论文研究目的，并且对论文所要研究的内容进行概述，对论文组织架构进行描述。

第2章研究基础。阐述本文的研究基础，主要介绍了协同过滤目前主要的三种方法，并对其具体的推荐实现过程进行了详细介绍，在本章最后对用户数据的收集方式进行了介绍。

第3章交互式协同过滤推荐方法的过程。本章主要分析了协同过滤目前主要的存在的问题，在此基础上提出本文的主要研究框架—交互式协同过滤推荐方法框架，并对框架的实现过程进行描述。最后对文中的用户交互行为进行定义，并对推荐列表的生成进行描述。

第4章基于用户评分与交互行为兴趣度的相似性计算方法。本章主要阐述了本文提出的基于用户评分差异度信息熵的相似性计算方法、基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，并在此基础上提出基于用户交互行

为兴趣度的相似性计算方法。

第5章基于推荐结果的用户交互行为兴趣度计算方法。本章主要阐述了基于用户项目的兴趣度计算方法以及基于用户类标签的兴趣度计算方法，最后通过兴趣度调整函数形成兴趣度矩阵。

第6章实验及结果分析。本章首先对实验需要的环境进行了说明，其次对数据集以及评价标准进行介绍，最后进行了两个对比实验验证本的URDE以及UIBS的有效性。

第7章总结与展望。本章对论文的主要的四个贡献点进行了总结，最后针对论文存在的问题进行了展望。

第 2 章 研究基础

本章阐述本文的研究基础，主要介绍了协同过滤目前主要的三种方法，并对其具体的推荐实现过程进行了详细介绍，在本章最后对用户数据的收集方式以及建模方法进行了介绍。

2.1 推荐技术

由于近三十年互联网的蓬勃发展，到今天，网络已经渗透大部分人的生活。而同时，相应的网络信息量也正在呈指数形式增长。推荐技术通过收集统计网络上用户或商品（包括普通商品、书籍、电影等）的信息，并对这些信息加以横向分析来学习预测用户的兴趣，据此产生特定的推荐方式，将预测结果推荐给用户。信息化时代的今天，效率和质量是每个人所追求的体验，成功的推荐技术在为用户提供服务的过程中，带来高质量的体验，更有可能帮助用户发现一些潜在的兴趣点和信息需求。由于上述优越性，推荐系统应运而生且应用广泛。

推荐系统的定义有很多，目前被广泛接受的定义是在上世纪末年由Resnick和Varian提出的[18]。目前，知名度较高的基于用户评分的MovieLens电影推荐系统，再如目前的一些电子商务网站京东淘宝等的推荐系统，通过分析用户的历史交易行为预测出买家的潜在购买商品，在网页上为该买家加以推荐。这两个实例，尤其是京东淘宝，不仅给用户提供了相较普通系统更为人性化的个性化服务体验，增加了用户的忠诚度，更因发展了很多潜在用户和交易，为电子商务领域创造了巨大的商业利益。

到目前为止，推荐技术的分类一直没有公认的统一标准。被广泛认可的主流推荐方法包括以下几种：基于内容的推荐技术[19]，基于关联规则的推荐技术[20]，基于协同过滤的推荐技术[21]以及混合推荐技术[22] 。其中基于协同过滤的推荐技术是应用最为广泛和成功的，也是本文采用的方法和研究重点。

协同过滤这一概念最早由Goldberg等人在1992年伴随其开发的应用邮件系统Tapestry 提出的[23]。其后美国明尼苏达大学计算机科学工程学院研究室在1996年开发出来的电影推荐系统GroupLens[24] 也是基于该技术的典型应用。协同过滤主要依据是：假设两个用户对于某些项目表现出共同的兴趣爱好或者是关注度（例如电影、书籍、音乐或者很多 其它形式商品），那么在其它项目上表现出共同兴趣的可能性也会很大。

对于判定两个用户之间是否有“共同兴趣爱好”的指标可以分为多种。可以通过用户

的历史交易浏览等行为去发现用户的潜在兴趣。通过这种方式进行数据的发现需要经过一系列的收集、分析，特别是分析过程要结合行为学和心理学来对数据进行深度挖掘，数据处理过程和数学模型的建立相对较为复杂。另外一种则是以用户对项目的评分（一般数值范围为0-5的整数，且分数越高表示兴趣程度越高）作为评判该用户对此项目喜好程度的依据。该方式因为项目评分是用户直接以数值形式给出，处理相对前者较为简单。

在把协同过滤算法应用到实际的推荐系统时，首先必须要建立协同过滤的数学模型，而量化这些评判标准的数学量称之为相似度。计算出相似度后再采取一定的筛选方法，为每个用户选取与之兴趣爱好更接近的用户集合，称之为用户近邻集，之后对用户进行预测评分从而产生推荐，其过程如图2.1所示。

图 2.1 协同过滤的一般过程

Fig 2.1 The general process of collaborative filtering

目前，协同过滤算法可以分为基于内存的协同过滤算法以及基于模型的协同过滤算法[25]，具体如图2.2所示。

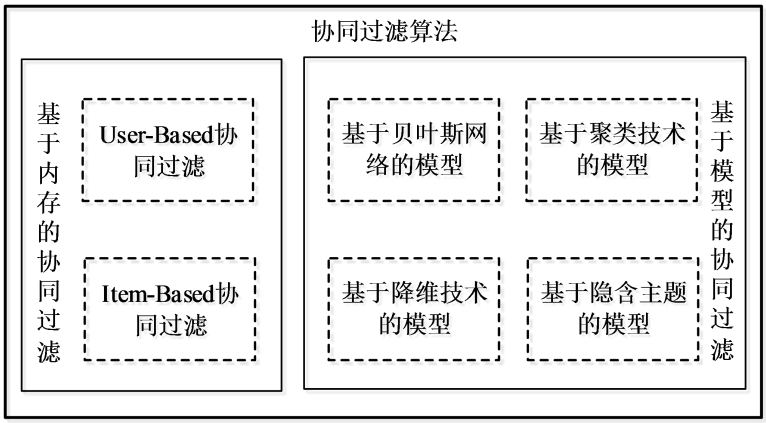

图 2.2 协同过滤推荐算法分类

Fig 2.1 Classification of collaborative filtering recommendation algorithms

2.1.1 基于用户的协同过滤推荐

基于用户的协同过滤[26]（User-based）推荐算法思想如图2.3所示。

User-based的协同过滤算法认为如果两个用户（或者买家）对部分项目（或者商品）的评分相近，即两者对这些项目的喜好程度相似，那么对其它项目的评分即感兴趣的程

度相似的可能性也很大[26]。该算法的基本步骤分为四步：用户兴趣模型的构建，用户相似性的计算，确定用户近邻集，预测评分及推荐。下面对此一一作简单介绍。

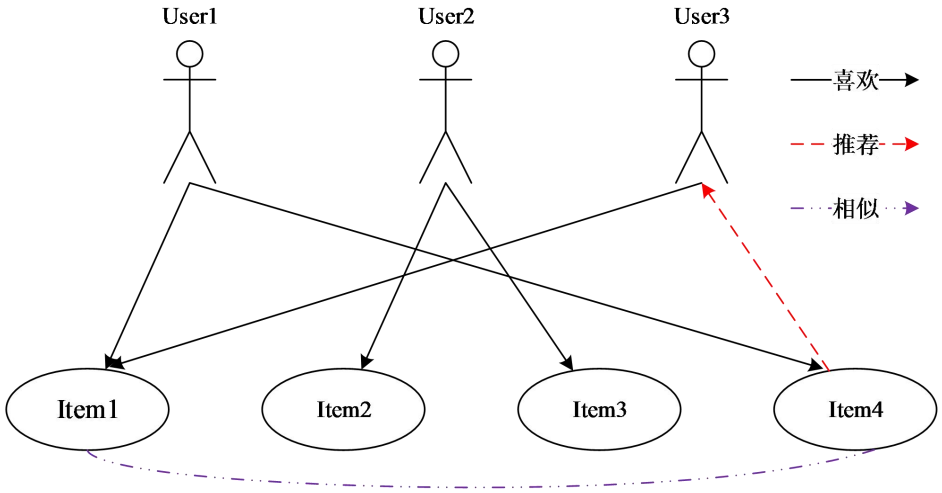

图 2.3 基于用户的协同过滤推荐

Fig. 2.3 User-based collaborative filtering recommendation

（1）用户兴趣模型的构建

假设已经从现有网络或者系统中收集到了用户和项目间的信息（用户对项目的评分）， 首先必须从这些数据中提取关键信息，并将其转换为合适的数学模型。对于该步骤，由用户评分行为转换的用户评分二维矩阵是被普遍采取的一种数学模型，如公式2.1所示。该矩阵 $R_{m\times n}$ 表示m个用户对n个项目的评价。其中评价信息可以有很多方法表示，比如两级评价标准、五级评价标准。

$$R_{m\times n}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{i1} & r_{i2} & \cdots & r_{in} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{2.1}$$

（2）用户相似性的计算

相似性的计算是协同过滤算法的核心，相似性的准确度直接影响着算法性能的好坏。较为传统的相似性计算方法有三种：Cosine (余弦）相似性计算，Pearson (皮尔森）相似性计算以及修正的余弦相似性计算[25]。

Cosine (余弦）相似性:设用户 i 和用户 j 在n维项目空间上的评分别用 $\vec{i}$ 和 $\vec{j}$ 两个向量表示，则用户 i 和 j 之间的相似度 $sim(i,j)$ 可以表示如公式2.2所示：

$$sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2\times\|\vec{j}\|_2}=\frac{\sum_{u\in U}R_{u,i}\cdot R_{u,j}}{\sqrt{\sum_{u\in U}R_{u,i}^2}\sqrt{\sum_{u\in U}R_{u,j}^2}} \tag{2.2}$$

Pearson（皮尔森）相似性：在该方法中，在对用户 i 和 j 求其相似性 $sim(i,j)$ 使，前提是知道 i 和 j 都进行评分过得集合 I_{ij}。Pearson相似性计算公式如2.3所示。

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-\overline{R}_i)(R_{j,c}-\overline{R}_j)}{\sqrt{\sum_{c\in I_{ij}}(R_{i,c}-\overline{R}_i)^2\sum_{c\in I_{ij}}(R_{j,c}-\overline{R}_j)^2}} \tag{2.3}$$

修正的余弦相似性:该方法认为用户的评分偏好是不同的，例如用户 i 认为很喜欢的项目评分值为5(5分制评分)，而相同情况下相同的体验用户 j 的评分值可能为4。造成这种评分值偏差的原因就是不同用户评分尺度的差异。该算法的设计主要是为了弥补余弦相似度算法中所忽略的不同用户的评分尺度不同的问题。计算公式如2.4所示。

$$sim(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\overline{R}_u)(R_{u,j}-\overline{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\overline{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\overline{R}_u)^2}} \tag{2.4}$$

（3）确定用户近邻集

用户近邻集的确定是仅次于相似度计算的协同过滤算法的又一个核心模块，近邻集搜索的质量和效率直接影响到了整个算法的准确性和有效性。常用的近邻集搜索方法有两种:

（a）K最近邻方法[27]

设定*k*表示用户近邻的大小。将除目标用户外的其它用户按照其与目标用户相似度值由大到小的顺序排序，选取前*k*个用户组成目标用户的近邻集。此方法实现简单，是目前最常用，被认为最成功的近邻搜索方法，如图2.4所示，其中*k*=7。

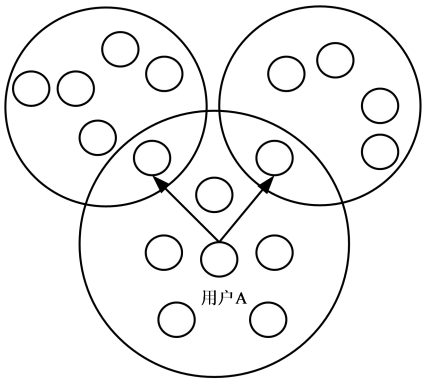

图 2.4 用户 A 的 k 个近邻

Fig. 2.4 The K neighbors of user A

（b）选择相似度大于阈值的用户

人为设定一个相似性阈值，依此作为划分近邻集的界定。其核心思想是：如果一个

用户与当前目标用户的相似性大于阈值，则将此用户划分到目标用户的近邻集。遍历除目标用户外的其它用户后，所形成的用户近邻集合就是目标用户的近邻集。此种方法的自由度较高，使用时可以根据实际数据的特点自行选取性能较好的阈值来划分近邻集，但是性能的判定标准比较复杂，不容易判断出好坏。

（4）预测评分及推荐

经过第三步得到目标用户的近邻集之后，下一步为目标用户u预测其未评分项目的评分值，并依此判定用户u对这些项目的喜欢的可能性。其方法如公式2.5所示。

$$P_{i,c} = \overline{R}_i + \frac{\sum_{m \in NN} sim(i,m) \cdot (R_{m,c} - \overline{R}_m)}{\sum_{m \in NN} (|sim(i,m)|)} \tag{2.5}$$

最后一步是分析预测结果，通过预测评分的对用户进行相关项目的推荐将。例如在5分制评分中，预测结果越接近5，则认为用户对项目喜欢的可能性越大；反之，预测结果越接近1则表示用户对项目不喜欢的可能性越大。

与近邻集的确定相类似的，为目标用户产生推荐同样有两种方法：一为设定一个阈值，若目标用户对某项目的预测评分值超过该阈值就将此项目推荐给用户，其余项目忽略；另一个是选取升序排序后的预测结果前N位推荐给用户，这种方法也被称为Top-N推荐。

基于用户的协同过滤被认为是最早也是比较经典的协同过滤算法，其优势在于依据用户平常的网页浏览模式和评分行为所表现出来的特点来发现不同用户的兴趣爱好。通过相似度的计算分析来挖掘有相似兴趣的用户，并据此进行推荐。这种以人为主体的数据挖掘方法有着更多的自由度以及更高的准确度。另一方面，此方法还能通过分析当前用户近邻的兴趣挖掘出目标用户的潜在兴趣，使系统更为人性化，同时将一些潜在用户发展成真正用户，为商家创造更多的利益。并且，用户兴趣模型直接依据用户评分数据建立，避免了用户兴趣模型学习的问题。这也是本文采取该方法的主要原因。

但是随着网络的发展，网络用户的数量以一种不可计量的方式在增加，使得该算法的复杂度越来越大。同时，用户数量的增大也使得用户评分矩阵极端稀疏，用户间共同评过分的项目偏少，进而影响到用户相似度计算的准确性，使得系统个性化推荐整体质量下降。

2.1.2 基于项目的协同过滤推荐

为了改进基于用户的协同过滤算法的局限性（即因用户数量过大而导致的预测准确度下降的问题），考虑到项目数量相较于用户数量增长速度较慢，数量上更为稳定，能提高效率，减小算法的复杂度。于是一个新的基于项目的协同过滤算法[29]被提出。

基于项目的协同过滤[28]（Item-based）推荐思想如图2.5所示。

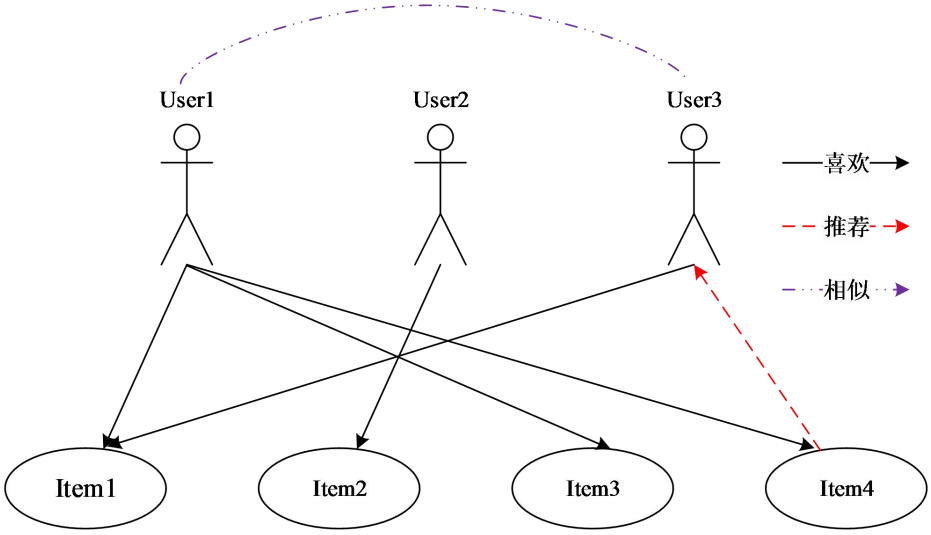

图 2.5 基于项目的协同过滤推荐

Fig. 2.5 Item-based collaborative filtering recommendation

从命名的方式可以看出，User-based和Item-based其算法思想一致，但是算法主体不同。Item-based的协同过滤算法所基于假设：如果大部分用户对某个项目有相似的态度（即某个项目同时被大部分用户所喜欢、讨厌或者认为一般），那么目标用户对该项目的态度也比较相似[27]。

基于项目协同过滤算法的基本过程与基于用户协同过滤算法的基本过程相类似。核心部分主要是项目间相似度的计算，项目邻居的搜索确定。而相似度的计算方法的分类和思想也是一致：也包括Cosine (余弦）相似度算法，Pearson (皮尔森）相似度算法以及修正的余弦相似性算法三种。最大的区别就是，两种方法的主体不同，基于用户的协同过滤算法的主体是人，而基于项目的协同过滤算法主体是项目。

2.1.3 基于模型的协同过滤推荐

由于协同过滤的推荐效果使其应用广泛，但是随着用户数量以及系统规模的扩大，增长迅速，评分矩阵稀疏问题日益突显，致使预测推荐效果受到影响。而且用户项目信息的剧增，也增加了算法处理数据的时间，减慢了系统运行速度，降低了系统推荐的效率。为了克服传统协同算法的这些缺陷，Breese等人提出基于概率的协同算法[30]。算法主要思想是：先根据用户的历史评分数据，运用数据统计、机器学习等方法建立用户模型。其预测评分公式描述如2.6所示。

$$P_{u,i} = E(R_{u,i}) = \sum_{i=0}^{m} pr(R_{u,k}, k \in R_u) \cdot i \tag{2.6}$$

其中 R_u 为目标用户u的评分集合。

该算法的数据基础是用户模型，建立用户模型常用的方法包括机器学习、Bayesian网络模型[31]、概率相关模型、最大熵模型等。

算法的核心部分是概率 *pr* 的计算，Breese提出两种概率选择模型[30]:贝叶斯网络模型和聚类模型。前者用贝叶斯网络中的每个节点表示每个项目，首先要学习用户的历史评分数据，得到用户兴趣模型，然后运用贝叶斯概率推荐规则计算目标用户对特定项目的预测评分。聚类模型是依据建立的用户兴趣模型，将用户（或项目）划分为不同的类，使同一类中的用户兴趣相近（或项目特征相类似）。在为目标用户预测时先确定好其所属的用户类别，再搜索该类别的用户空间（或依据用户兴趣，在特定的项目类别中搜索），有效减小了搜索空间，提高了效率。这种方法虽然实时性效果较好，但由于实际问题中模型建立的难度以及数据量的大小使得基于模型的协同过滤方法使用范围并不是很大。

2.2 用户建模

在用户建模的过程中，对用户偏好获取是至关重要的，模型的建立过程可分为用户数据收集和兴趣模型阶段，以下便是对这两阶段的具体分析。

2.2.1 用户数据收集

（1）显式数据收集

显式数据是用户在与系统的交互过程中的直观数据，是用户主动向系统提供的数据。例如用户对项目的评分，用户在使用系统时注册填写的信息等等。一般来说，这种数据的价值比较大，它直接反映了用户对某类或某种项目的偏好，这种数在协同过滤算法推荐的过程中提取难度相对较小，最后推荐的质量相对较高。但同时这种数据需要用户的高度参与，降低了用户体验，因此获取难度比较大。

目前，常见显式数据收集用户兴趣的方式有：要求用户评价其浏览的项目；要求用户回答预先设定的问题，从而在其中收集到用户的兴趣偏好。前者虽然较为简单，也能及时了解到用户的兴趣，但是常存在稀疏数据的问题；后者通常可以得到一个较为完整的用户兴趣模板，但是存在无法跟踪和了解用户兴趣的变化的问题。

（2）隐式数据收集

隐式数据通常是指在不需要用户的主动参与并且对用户透明的方式下间接获取用户的各种活动信息。比如用户网页浏览的链接、历史交易的记录、页面停留的时间、浏览过程中所点击的区域、拖动滚动条次数等都属于隐式数据采集的范围。相较于显示数据，隐式数据所包含的数据内容和含义都更加丰富。

相对于显式数据，隐式数据的收集方式相对比较困难，其中也包含了大量噪声。目前常用的方式主要通过用户的各种日志数据进行提取。之后通过各种人工智能技术对日志数据进行分析，发现其中的隐含用户兴趣，从而更好地对其进行使用。

2.2.2 用户模型表示

用户项目的评分矩阵模型：

用户项目评分矩阵是协同过滤中表示用户项目关系最常用的方法[32] [33]。该模型如公式2.7所示，通过一个矩阵表示用户对项目的评分，由于采用的评分制不同，因此整个的评分范围也不同，但通常评分值越大表明用户对项目的偏好程度越大。

$$R_{m\times n}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{i1} & r_{i2} & \cdots & r_{in} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{2.7}$$

用户评分矩阵模型相对比较容易，矩阵的形成相对比较简单，因此也适用于各种相似性计算方法，具有很好。但是，在一个系统中由于用户数目和项目数很大，会造成用户评分矩阵的稀疏，因此这也是协同过滤亟待解决的问题。

2.3 本章小结

本章阐述本文的相关研究，本文研究所涉及的内容主要包括协同过滤目前主要的三种方法：基于用户的协同过滤推荐、基于项目的协同过滤推荐以及基于模型的协同过滤推荐，重点对相似性计算方法以及用户近邻的选择方法进行介绍，最后对用户数据的获取以及用户模型进行介绍。

第 3 章 交互式协同过滤推荐过程

本章主要描述交互式协同过滤推荐方法的基本过程。首先详细分析了目前协同过滤中存在的问题，基于目前协同过滤中存在的数据稀疏性以及用户推荐反馈等问题，提出交互式协同过滤推荐方法的框架以及交互式协同过滤推荐方法的处理过程。其次在研究框架的基础上对用户交互行为的进行了详细的分析，同时对协同过滤推荐列表的生成方法进行了阐述。

3.1 问题的提出

目前，协同过滤在推荐系统中取得的效果非常明显，大量的实践已经证明了其在推荐领域的价值。协同过滤是针对用户之间偏好进行推荐的，算法认为具有相同习惯、兴趣的用户，其需求信息在很大程度上也存在相似性，基于这种思想对信息进行过滤，进而推荐给用户。协同过滤相对于其它的推荐算法，能够更有效的利用用户历史评价数据，从而可以在用户反馈较少的情况下进行快速学习推荐。

对于一个大型的系统，每个用户能够接触到的商品是比较少的，用户评价数据更少。例如，中国著名的网上购物平台—淘宝的商品非常多，但是正常用户涉及到的商品也就其中的1%-2%，甚至更少，这就造成了用户评分矩阵的稀疏性问题，难以通过寻找用户近邻进行推荐，其推荐效果大大降低[34]。同时，对于一个大型的推荐系统来说，用户在进行相关项目选择的过程中会不断地与系统进行交互，产生了大量的用户交互行为数据，这些交互行为数据中蕴含这丰富的用户兴趣信息，对其进行收集、分析、进而进行利用，这个以及是非常大的。

针对上述分析，本文主要对协同过滤存在的以下问题进行研究：

（1）稀疏性问题。

传统的相似性计算方法针对高维数据的稀疏性问题，由于用户的共同评价数据的规模大小不一，且用户自身评分偏好的不同，得出的用户间相似性和真实值存在一定偏差，往往不理想。因此在这种情况下需要考虑评分矩阵的差异化，考虑用户自身的评分偏好以及用户间的评分差异。

（2）用户推荐反馈问题

目前的推荐大都是一种静态的推荐，即根据用户的评分矩阵进行计算以推荐给用户新的可能感兴趣的项目。但是，推荐是一个交互式的动态过程。用户对推荐列表中物品的兴趣度将会反映用户兴趣度的变化，即用户对推荐结果的反馈，应该作为推荐列表更

新优化的依据。为此如何解决用户的推荐反馈是一个关键问题。

（3）基于用户反馈的推荐优化问题

在用户推荐反馈得到解决之后，如何将反馈应用到整个推荐的过程中从而优化推荐成为一个关键问题。

3.2 交互式协同过滤推荐方法的处理过程

针对3.1节中本文拟解决的三个关键问题，本文设计了交互式协同过滤推荐的基本框架。根据该基本框架，结合实验方法，给出交互式协同过滤推荐方法的具体实现过程。下面具体说明。

3.2.1 交互式协同过滤推荐方法框架

针对数据稀疏性问题、用户推荐反馈问题以及基于用户反馈的推荐优化问题，本文提出交互式协同过滤推荐方法。

该方法是一个对协同过滤优化的过程。通过处理原始的用户评分数据，得到用户-项目评分矩阵，此时评分矩阵往往存在稀疏性问题，同时由于用户间评分的差异，用户评分矩阵的数据也参差不齐，针对这个问题，本文提出基于用户评分差异度信息熵的相似性计算方法。该方法通过衡量用户间评分的差异性，引入信息熵来进行用户相似性的计算，在此基础上生成推荐列表；针对用户的推荐反馈的问题，通过考虑用户不同交互行为包含用户兴趣信息不同，引入兴趣度的概念对其进行衡量，通过不同的兴趣度计算方法分别计算用户项目兴趣度矩阵以及用户类标签兴趣度矩阵；针对基于用户反馈的推荐优化问题，本文提出基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法将用户兴趣度矩阵作用到协同过滤的推荐过程中，优化推荐。交互式协同过滤推荐框架如图3.1所示。

由图3.1可知，交互式协同过滤推荐方法可以分为以下几个问题：

（1）基于用户评分差异度信息熵的相似性计算方法

相似性计算在协同过滤推荐算法中一直是一个关键问题，但是以往的算法中往往面临着评分数据稀疏的问题，这样用户之间的相似性往往会比实际的结果偏大或者偏小，本文在初始的协同过滤推荐过程中，提出一个基于用户评分差异度信息熵的相似性计算方法，主要对用户的评分偏好进行差异化处理，同时通过信息熵的方法衡量用户评分的相似性，进而生成推荐列表，为后续相似度的计算提供基础。

（2）基于推荐结果的用户交互行为兴趣度计算

本文的交互行为主要是用户对推荐结果的操作，不同的用户交互行为反映了用户不

同的兴趣度。本文将用户的行为分为成基于项目的行为以及基于类标签的行为：基于项目的行为主要是指用户与推荐列表之间的交互行为，在此基础上通过计算基于项目的兴趣度形成用户兴趣度矩阵；基于类标签的行为主要是指用户与分类标签之间的交互行为，在此基础上通过计算基于类标签的兴趣度形成用户兴趣度矩阵。

（3）基于用户交互行为兴趣度的相似性计算方法

基于用户交互行为兴趣度的相似性计算方法主要分为基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，分别针对用户项目兴趣度矩阵和用户类标签兴趣度矩阵进行计算得到对应的用户相似性，计算得到这两种相似性，通过基于用户交互行为兴趣度的相似性计算方法来进行相似性的计算。

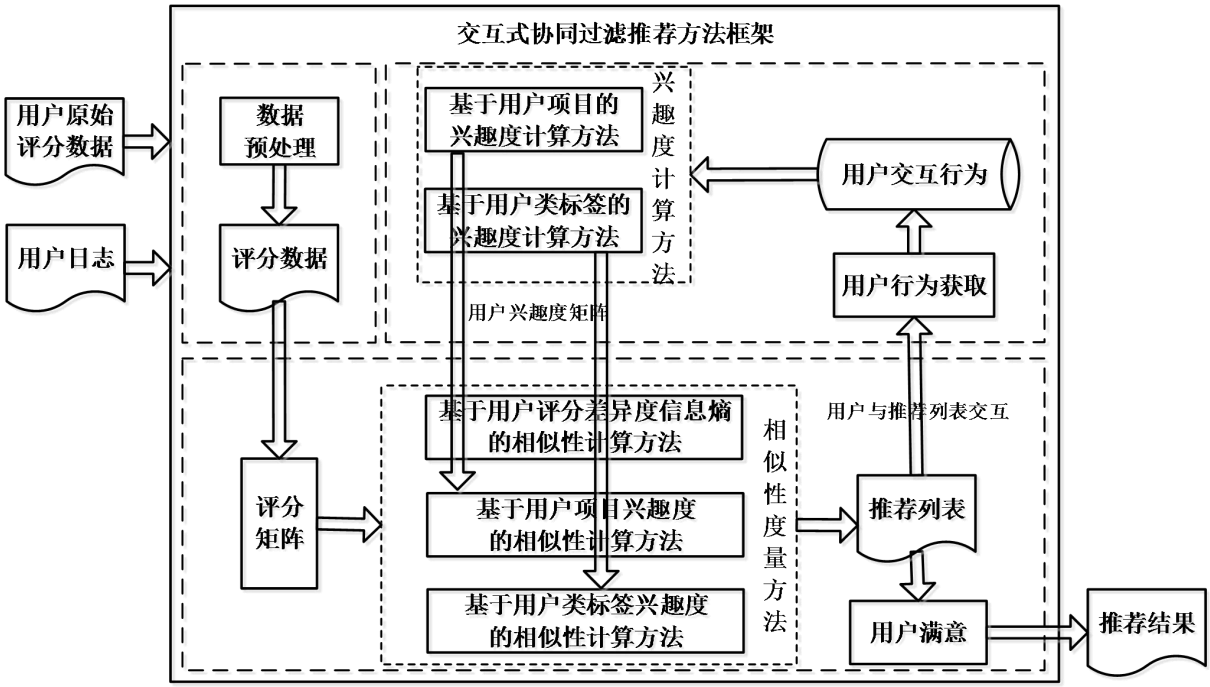

图 3.1 交互式协同过滤推荐框架

Fig. 3.1 The frame of interactive collaborative filtering recommendation

3.2.2 交互式协同过滤实现过程

基于3.2.1节中的交互式协同过滤推荐方法框架，本文给出了交互式协同过滤推荐方法的实现过程，具体如图3.2所示。实现过程分为四个阶段。

（1）用户原始数据处理阶段。

以电影推荐为例，针对用户的初始评分，这个阶段主要需要解决两个关键问题：第一个是用户评分数据的选取问题，由于的用户评分数据中不仅仅是用户的评分信息，还可能包含影片的标签信息，评分时间信息等等，这里需要选取用户的标识，被评分电影的标识以及用户对电影的评分信息，影片标签信息等等；第二个问题是用户评分矩阵的

生成，用户的评分信息主要是服务于协同过滤的初始用户推荐问题，因此这里需要把数据转换成协同过滤需要的用户-项目评分矩阵。

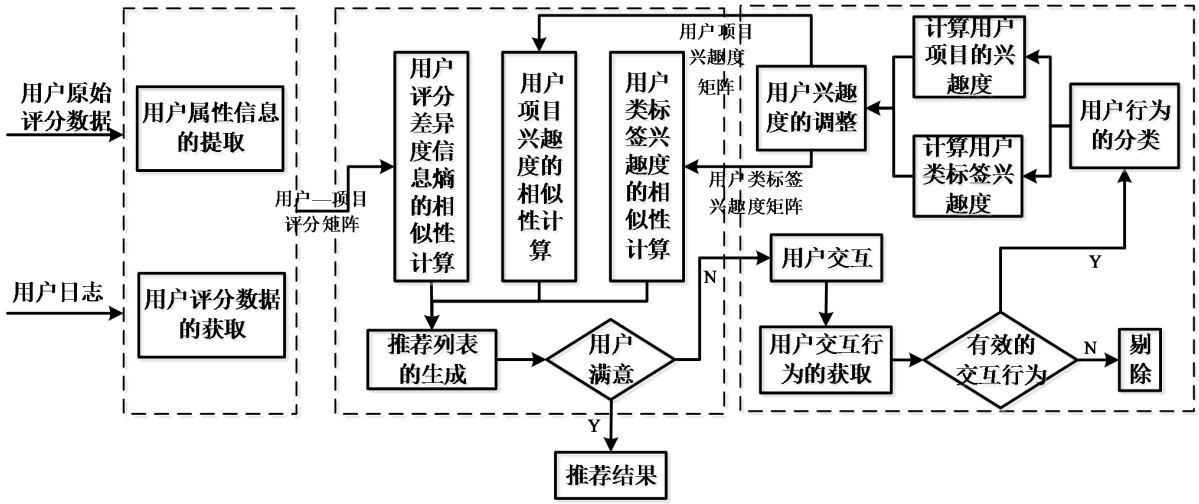

图 3.2 交互式协同过滤推荐实现过程

Fig. 3.2 The implementation process of interactive collaborative filtering recommendation

（2）基于用户评分差异度信息熵的相似性计算阶段

这里需要对步骤（1）得出的用户-项目评分矩阵进行处理从而得到推荐列表。用户评分矩阵存在的数据稀疏性问题会对最终的推荐结果产生很大的影响，本文主要考虑了不同用户评分之间的差异性以及偏好性，通过对用户评分数据的差异化处理，进而通过信息熵的方法来衡量用户之间差异程度，计算用户之间相似性，通过用户近邻的选择从而生成推荐列表。

（3）基于用户推荐结果的交互行为兴趣度计算阶段

阶段（2）得到的推荐列表并不一定是用户期望的结果，通过用户对推荐列表的交互式行为，来进行个人兴趣的调整，这里的用户交互行为主要分为用户对推荐列表的交互行为以及用户对类标签的交互行为。针对用户对类标签的交互式行为，给出基于用户类标签的兴趣度计算方法，形成用户类标签兴趣度矩阵，反映了用户对类标签的感兴趣程度；针对用户对推荐列表的交互式行为，给出基于用户项目的兴趣度计算方法，形成用户项目兴趣度矩阵，反映了用户对项目的感兴趣程度；

（4）基于用户交互行为兴趣度的相似性计算阶段

在阶段（3）的基础上本文给出了基于用户交互行为兴趣度的相似性计算，该方法包含两个部分：基于用户类标签兴趣度的相似性计算以及基于用户项目兴趣度的相似性计算。其中，基于用户类标签的兴趣度的相似性计算，在目前pearson相似性计算的基础上，通过引入用户类标签兴趣度，进而计算出用户之间的相似性。基于用户项目兴趣度

的相似性计算，在目前pearson相似性计算的基础上，通过引入用户项目兴趣度，进而计算出用户之间的相似性，通过调整两种相似性计算方法的权重来得到推荐反馈后的用户相似性，得到优化后的推荐结果。

（5）用户兴趣度的更新以及推荐的评价

由于用户的兴趣会随着时间的推移出现“兴趣漂移”的现象，因此需要对用户兴趣度矩阵进行更新，从而更好的服务于协同过滤推荐方法。本文通过滑动时间窗口的方法进行用户兴趣度的更新，主要考虑了时间因素。同时通过对推荐结果性能的评价作为用户满意度的衡量标准，达到一定标准给出推荐列表，推荐结束。

3.3节详细介绍了本文采用的交互式协同过滤推荐方法中用户交互行为的定义以及获取方法。

3.3 基于推荐结果的用户交互行为

为了便于分析和归纳用户交互行为特征，本文将用户交互行为分为用户的主动交互行为和用户被动交互行为。主动交互行为是指用户主动调整用户信息，这里主要是用户鼠标拖动行为；被动交互行为是指用户参与交互过程，但是用户并不是直接表达个人兴趣，而是通过交互行为去获取用户兴趣度，这里主要是指用户对项目的访问行为、用户对项目的影片观看行为、用户对项目的鼠标操作行为。这里对其进行了定义。

为了便于定义基于推荐结果的用户交互行为这里给出推荐列表条目以及列表条目结果页面的定义。

【定义3.2】推荐列表条目（Recommended list Item）：通过协同过滤算法得到推荐列表 $Item =< Item_1, Item_2, ..., Item_m >$，表示用户感兴趣的项目，其中每一项 $Item_i$ 代表一个推荐列表条目。这里使用一个二元组对用户交互行为模型进行形式化定义：*RLI=<U,Item>*，其中：

- $U = \{u_1, u_2, ..., u_m\}$ *为用户集合；*
- $Item = \{Item_1, Item_2, ..., Item_m\}$ *为推荐列表条目的集合；*

【定义 3.2】推荐列表条目结果页面（Recommended Item Page）：这里使用一个二元组对用户交互行为模型进行形式化定义：*RIP=<U,Item,Page>*，其中：

- $U = \{u_1, u_2, ..., u_m\}$ *为用户集合；*
- $Item = \{Item_1, Item_2, ..., Item_m\}$ *为推荐列表条目的集合；*
- *Page 为 Item 对应的页面；*

【定义3.3】用户对类标签的鼠标拖动行为UMDB（User Mouse Drag Behavior for

Category Lable）：用户通过拖动鼠标使其靠近其感兴趣的标签，通过这种交互式行为来直接表达个人兴趣。用户对类标签的鼠标拖动行为 UMDB 形式化定义为：*UMDB=<BeID,U,Lab,LLab,LA,LB>*，其中：

- *BeID* 为用户交互行为的唯一标识；
- $U=\{u_1,u_2,...,u_m\}$ 为用户集合；
- *Lab* 为分类标签；
- *LLab* 为用户标签位置；
- *LA* 为用户拖动后的位置；
- *LB* 为用户初始位置；

【定义 3.4】用户对项目的访问行为 UAIB（User Access Recommended Item Behavior）：用户对其感兴趣的推荐项目会进行访问行为，因此通过用户的访问行为去反推用户的兴趣是可行的。这里主要记录用户的访问次数，用户每次点击推荐列表条目都会被记录 1 次。用户多次点击推荐列表条目，点击量值递增，即推荐列表条目的点击次数。用户对项目的访问行为 UAIB 可以被形式化定义为：*UAIB=<BeID,U,Item,MID,Fre>*，其中：

- BeID 为用户交互式行为的唯一标识；
- $U=\{u_1,u_2,...,u_m\}$ 为用户集合；
- $Item=\{Item_1,Item_2,...,Item_m\}$ 为推荐列表条目的集合；
- *MID* 为推荐列表条目对应项目页面 *Page*；
- *Fre* 为推荐列表条目的访问次数，每访问一次，*Fre* 的值会增加 1。

【定义 3.5】用户对项目的影片观看行为 UWMB（User Watch Movie Behavior for Iterm）：本文主要针对 MovieLens 电影数据进行分析处理。用户对其感兴趣的推荐列表条目会进行点击浏览，如果兴趣比较大，用户会对其产生观看行为，但是如果仅仅考虑用户的观看时长，而忽略了影片本身的长度，会造成用户兴趣评判的不准确，因为观看的时长会受到电影时长的影响，在本文使用的是用户观看时长占比的概念。用户观看时长占比越大说明用户对其越感兴趣，反之，兴趣比较小。用户对项目的影片观看行为 UWMB 形式化定义为：*UWMB=<BeID,U,MID,MT,WT>*，其中：

- BeID 为用户交互式行为的唯一标识；
- $U=\{u_1,u_2,...,u_m\}$ 为用户集合；
- *MID* 为推荐列表条目对应项目页面 *Page*；
- MT 为影片本身长度；

- WT 为用户观看时长
- 这里定义用户观看时长占比为：$MPro = WT / MT$，分析可知，$MPro$ 的范围为[0.1]。

【定义 3.6】用户对项目的鼠标操作行为 UOMB（User Operate Mouse Behavior for Item）：用户的兴趣可以通过其鼠标的操作行为反映，比如用户对其喜欢的推荐项目会进行各种鼠标、键盘的操作，包括鼠标左键、右键操作，键盘的 PageUp、PageDown 操作，以及光标的操作等等。用户对项目的鼠标操作行为 UOMB 形式化定义为：$UOMB=<BeID,U,Item,ML,MR,PD,PU,CC,MC>$，其中：

- $BeID$ 为用户交互式行为的唯一标识；
- $U=\{u_1,u_2,...,u_m\}$ 为用户集合；
- $Item=\{Item_1,Item_2,...,Item_m\}$ 为推荐列表条目的集合；
- MID 为推荐列表条目对应项目页面 $Page$；
- ML 为鼠标左键点击次数；
- MR 为鼠标右键操作次数；
- PD 为按下 PageDown 的次数；
- PU 为按下 PageUp 的次数；
- CC 为光标键在产生的滚动次数；
- MC 为鼠标滚轮产生的滚动次数。

3.4 推荐列表生成

推荐算法的目的是生成推荐结果，这里定义为推荐列表，其计算公式如3.1所示。同时本文提出的交互式协同过滤推荐的交互过程也是针对最终的推荐结果列表进行的相关交互行为，通过交互行为计算用户的兴趣信息，作用于用户性相似性的计算，从而优化推荐列表

在得到用户相似性 $Sim(u_i,u_j)$ 之后，就可以通过公式3.1对其进行评分预测，具体如3.1所示。

$$P_{a,i}=\overline{R_a}+\frac{\sum_{u\in KNB} Sim(u_i,u_j)\times(R_{u,i}-\overline{R_u})}{\sum_{u\in KNB}(Sim(u_i,u_j))} \tag{3.1}$$

- $P_{a,i}$ 代表用户 u_i 对于项目 i 的预测评分；
- KNB 表示目标用户 u_i 的近邻集合；
- $\overline{R_a}$ 和 $\overline{R_u}$ 代表用户 u_i 在所有已评分项目上的平均评分；
- $R_{u,i}$ 表示用户 u_i 对项目 i 的评分值；

最后对评分预测值以Top-N的方法提取N个项目作为推荐结果。

3.5 本章小结

本章主要分析了协同过滤目前主要的存在的问题，在此基础上提出本文的主要研究框架-交互式协同过滤推荐方法框架，并对框架的实现过程进行描述。最后对文中的用户交互行为详细说明，对推荐列表的生成进行描述，以此作为第四章、第五章的研究基础。

第 4 章 基于用户评分与交互行为兴趣度的相似性计算方法

精确计算用户之间的相似性可以帮助更准确对用户进行近邻的计算，进而提高最终的推荐质量。由于传统协同过滤推荐算法中相似性的计算往往依赖于用户评分数据，因此评分数据的稀疏性问题以及个人评分喜好的不同会对最终的结果产生很大影响，同时考虑用户交互行为对用户最终推荐结果的影响，本章通过分析对现有相似性计算方法的分析，提出基于用户评分与交互行为的相似性计算方法。

4.1 相关工作

在协同过滤算法中数据稀疏性对最终的推荐精度影响很大。数据稀疏性主要由于用户评分矩阵用户数、项目数过多，但是用户真正能参与的评分项目有限，这样就造成了评分矩阵的稀疏性问题。

目前解决这一问题的主要方式是对稀疏性矩阵进行数据填充 [35]，常用的方法是对于用户没有评分的项目以0值进行填充，但是这种方式由于主观性较强，完全以一个数值代替缺失值，会对最终的推荐结果造成很大影响。数据填充在一定程度上缓解了这一问题，本质上并没有解决。除此之外，将评分矩阵结合内容加入到整个推荐系统中的方法也被研究并取得了一定效果。

通过协同过滤的数据稀疏性问题，本文提出基于用户评分与交互行为兴趣度的相似性计算方法，该方法对目前相似性计算进行了扩展。考虑的用户间评分的差异性，同时考虑用户本身评分的偏好，本文提出一种基于用户评分差异度信息熵的相似性计算方法来衡量用户评分的相似程度。考虑用户交互行为推荐反馈的问题，本文提出基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，分别对应用户的两种不同兴趣度矩阵。在此基础上，通过基于用户交互行为兴趣度的相似计算方法中权重的调整，解决用户推荐反馈问题。该方法的具体细节将在下文中进行介绍。

4.2 基于用户评分与交互行为兴趣度的相似性计算方法基本思路

本节主要是对本章提出的基于用户评分与交互行为兴趣度的相似性计算方法基本处理过程进行描述。

4.2.1 问题描述

在协同过滤推荐的过程中，相似性计算一直都是整个算法的核心，合适的相似性计算方法会对最终的推荐结果很大影响，目前关于相似性计算的研究比较多，传统的方法有Pearson相似性计算方法、Spearman相似性计算方法、余弦相似性计算方法、修正的余弦相似性计算方法等等。虽然这些方法在一定环境中能够各自取得很好的推荐效果，但仍有共同的问题亟待解决。

传统的相似性计算方法针对高维数据的稀疏性问题，由于用户的共同评价数据的规模大小不一，且用户自身评分偏好的不同，得出的用户间相似性和真实值存在一定偏差，往往不理想。因此在这种情况下需要对用户自身的评分偏好、用户间评分的差异程度以及用户共同评分的项目规模进行考虑。

4.2.2 问题解决的基本思路

通过4.1节中相关工作的分析，结合交互式推荐中的具体问题，本文提出于用户评分与交互行为兴趣度的相似性计算方法，此方法的处理过程如图4.1所示。

首先通过用户评分数据形成用户-项目评分矩阵，接下来计算用户间差异度，计算用户加权差异度信息熵，形成用户相似性矩阵，用于推荐。针对推荐结果的好坏，若用户满意，给出此时的推荐结果，如果用户不满意，通过用户交互计算用户不同交互行为对应的兴趣度矩阵，之后针对不同的兴趣度矩阵分别使用基于用户项目兴趣度的相似性计算方法，基于用户类标签兴趣度的相似性计算方法分别计算用户相似性，之后通过调整两种相似性计算方法的权重重新进行推荐，重复上述步骤，直至用户满意。

这里包含三个主要问题：

（1）基于用户评分差异度信息熵的相似性计算方法。

传统的相似性计算方法针对高维数据的稀疏性问题，由于用户的共同评价数据的规模大小不一，且用户自身评分偏好的不同，得出的用户间相似性和真实值存在一定偏差，会影响最终推荐结果的质量。因此在这种情况下需要考虑评分矩阵的差异化，考虑用户自身的评分偏好以及用户间的评分差异，这里对基于用户评分差异度信息熵的相似性计算方法进行研究。

（2）基于用户项目兴趣度的相似性计算方法。

该方法一种优化推荐的方法，通过协同过滤推荐方法给出的推荐列表，用户通过交互式的行为去表达个人喜好，比如用户的鼠标操作行为，用户的观看行为，用户的列表条目点击行为，这些行为反应了用户兴趣，通过用户兴趣度的计算，生成用户兴趣度矩阵，来进行用户相似性的计算，进而优化推荐结果。这里对基于用户项目兴趣度的相似性计算方法进行研究。

（3）基于用户类标签兴趣度的相似性计算方法。

该方法这也是一种优化推荐的方法，这是用于解决用户数据稀疏性的方法，用户不要大量数据，用户通过交互式的行为调整用户的喜好标签，通过对用户喜好标签兴趣都的计算生成用户兴趣度矩阵，来进行用户相似性的计算，进而优化推荐结果。这里对基于用户类标签兴趣度的相似性计算方法进行研究。

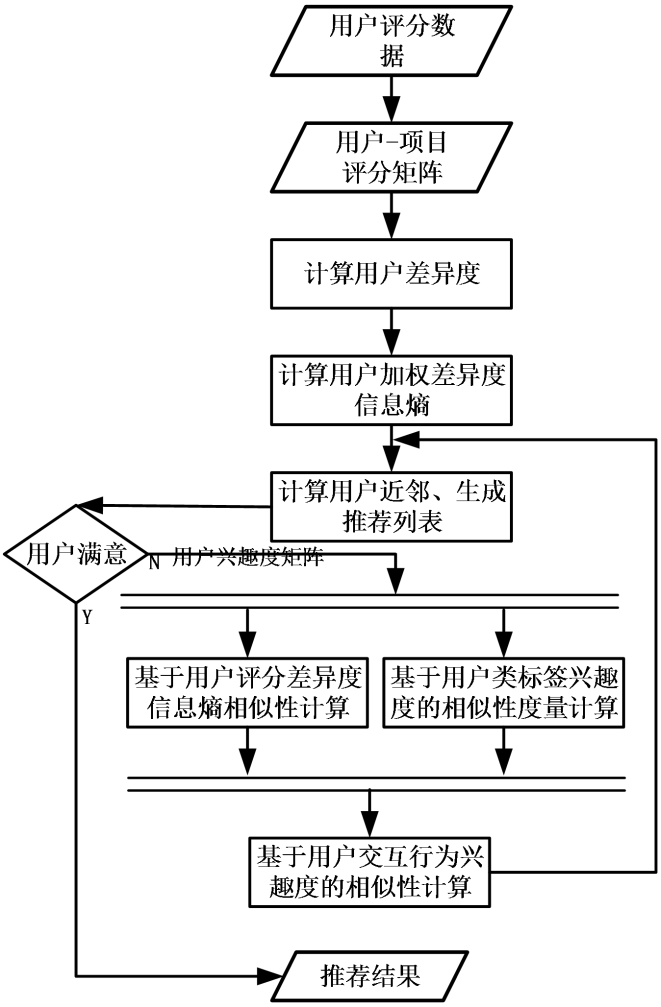

图 4.1 基于用户评分与交互行为兴趣度的相似性计算方法过程图

Fig. 4.1 The process of user's Similarity calculation method based on user rating and interactive behavior interest degree

4.3 基于用户评分与交互行为的相似性计算方法

为了更好对本章进行研究，这里给出兴趣度定义如定义4.1。

【定义4.1】兴趣度(Interest Degree)：兴趣度用于衡量用户对项目的感兴趣程度。这里把它分为基于用户项目的兴趣度，基于用户类标签的兴趣度，分别用于本章中对应的相似性计算。

其中基于用户项目的兴趣度主要是通过用户对推荐列表条目的交互式行为，衡量用

户对其兴趣，这里使用IDI对其进行表示，最后通过计算形成用户兴趣度矩阵。

基于用户类标签的兴趣度主要是通过用户交互式调整其与不同类别标签的距离，表达其对相应标签的兴趣程度，这里使用IDL对其进行表示，最后通过计算形成用户兴趣度矩阵。

为了更好地对本节的相似性计算方法进行表示，这里给出用户评分与交互行为兴趣度相似性模型的定义如定义4.2所示。

【定义4.2】用户评分与交互行为兴趣相似性模型 (User Rating and Interactive Behavior Interest Similarity Model)：考虑到具体的协同过滤中相似度计算的问题，本文提出用户评分与交互行为兴趣相似性模型的概念，该模型形式化定义为一个三元组 $IM =< URDE, ITSim, ILSim >$，将相似度计算分为三个部分，其中：

- $URDE$ 代表基于用户评分差异度信息熵的相似性。
- $ITSim$ 代表基于用户项目兴趣度的相似性。
- $ILSim$ 代表基于用户类标签兴趣度的相似性。

下面章节分别对这三种相似性计算方法进行研究。

4.3.1 基于用户评分差异度信息熵的相似性计算方法

（1）基于用户评分差异度信息熵的相似性计算方法基本思想

在协同过滤传统的相似性计算中，由于对用户自身评分偏好的不同、用户间的评分差异、用户间共同评分规模的考虑不足，使得相似性精度在一定程度上受到影响。本节在这些问题的基础上，对评分数据的混乱程度以及分散程度进一步考虑，引入差熵的方法来进行解决。

信息熵[36]作为衡量数据分布离散程度的一种度量，分布越离散，其熵值越大；分布越有序，其熵值越小。在具体的推荐问题中，对于用户评分数据集 $R_{m\times n}$，其公式如4.1所示。

$$H(R)=\sum_{i=1}^{n} p(r_i)\log_2\frac{1}{p(r_i)} \tag{4.1}$$

在公式4.1中，n表示评分数据集中的分类数，比如MovieLens数据集就是五级评分标准，那么这里的分类数对应的就是5。$p(r_i)$ 是类别对应的概率。$H(R)$ 代表熵值，衡量数据的离散程度。

根据上述分析，可以将信息熵用于衡量用户之间评分差异的离散程度，例如，对于MovieLens数据集，用户评分数据 $R_{ik}\in[1,5], R_{jk}\in[1,5]$，通过计算用户评分的差异度，得出（$R_{ik}$-$R_{jk}$）$\in$[-4,4]，共分为9个类别，分别统计各自出现次数，进而计算概率值。因此

这里将信息熵应用到协同过滤算法中的相似性计算中。

除此之外还需要考虑用户自身的评分偏好，比如针对MovieLens数据集，有些用户偏好于使用1到5来对影片进行评分，而有些用户更倾向于使用2到3或者3到4；另一方面不同用户的评分尺度也往往存在差异，比如有些用户习惯用3或者4去表示喜欢，对于评分5的使用更谨慎，有些用户则直接使用5。这就需要对不同用户的评分数据进行差异化处理。在此基础上，考虑用户的共同评分项目集对计算的影响，如果用户A与用户B的共同评分项目比较一致，那么互相之间的潜在影响比较大，因此在相似性计算的过程中需要进行考虑。

针对上述分析本文提出基于用户评分差异度信息熵的相似性计算方法（URDE）用于计算用户间的相似性。

（2）基于用户评分差异度信息熵的相似性计算方法详细过程

URDE方法主要是通过差异度来消除不同用户的不同评分范围以及评分尺度的影响，通过信息熵来衡量用户间评分的差异程度。最后通过加权处理，归一化处理得到用户间相似性。

基于差异度信息熵的相似性计算方法详细过程如图4.2所示。

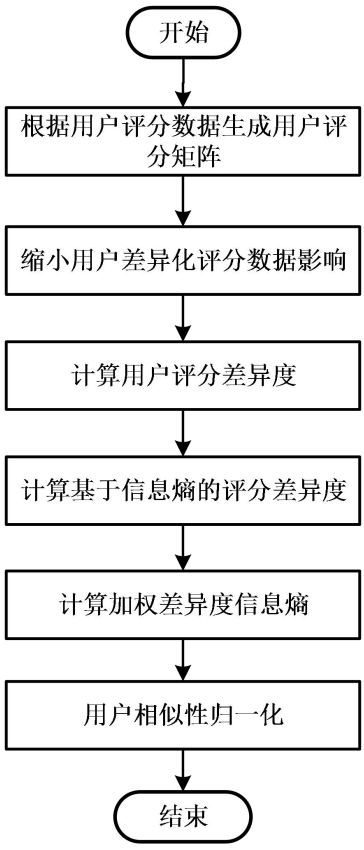

图 4.2 基于用户评分差异度信息熵的相似性计算方法过程图

Fig. 4.2 The process of computing user’s conceptual understanding ability

URDE方法通过四个步骤完成：

Step 1：消除用户自身评分偏好的影响

用户不同的评分尺度以及评分偏好会影响用户的评分数据，因此这里需要对其进行处理，减小其带来的影响，其公式如4.2、4.3所示。

$$r_{ik}=(r_{ik}-\bar{r}_i+\delta),r_{jk}=(r_{jk}-\bar{r}_j+\delta) \tag{4.2}$$

$$r_{ik}=(r_{ik}-min(r_i)),r_{jk}=(r_{jk}-min(r_i)) \tag{4.3}$$

其中，r_{ik}和r_{jk}分别表示用户i和用户j对项目k的评分值，δ和$min(r_i)$是调节参数使r_{ik}评分介于[1,5]。

Step 2：计算用户间评分差异度

协同过滤主要针对的是用户评分数据，但是，不同用户间的评分不同，在消除用户自身评分偏好和尺度的同时也需要对用户间的评分差异程度进行处理，以减小其带来的影响。

其中，用户共同评分项目对应下标如公式4.4所示。

$$\begin{aligned} SaRU(u_i,u_j)&=\{1,...,k,...,l\} \\ l&=len(SaRU(u_i,u_j)) \end{aligned} \tag{4.4}$$

$DR(u_i,u_j)$是用户共同评分的差异值集合，dr_k是用户i和用户j对项目k的评分差异值。如公式4.5所示。

$$\begin{aligned} DR(u_i,u_j)&=\{r_{i,1}\text{-}r_{j,1},...,r_{i,k}\text{-}r_{j,k},...,r_{i,n}\text{-}r_{j,n}\},k\in SaRU \\ &=\{dr_1,...,dr_k,...,dr_n\} \end{aligned} \tag{4.5}$$

Step 3：计算加权差异度信息熵

通过引入信息熵理论用于衡量用户间评分差异的离散程度，这里通过公式4.6计算得到用户差异度信息熵。其中，$p(dr_i)$代表dr_i在$DR(u_i,u_j)$中的概率大小。通过信息熵公式4.6可知，如果所有的$fre(dr_k)$相同，熵值为最大。如果dr_1=dr_k=dr_n均相同，此时熵值为最小。

$$\begin{aligned} H(DR(u_i,u_j))&=\sum_{i=1}^{n}p(dr_i)log_2\frac{1}{p(dr_i)} \\ p(dr_i)&=\frac{fre(dr_i)}{\sum_{k=1}^{L}fre(dr_k)} \end{aligned} \tag{4.6}$$

由于各级评分差值反映的用户间相似程度不同，因此通过公式4.6进行差异度信息熵

的计算存在一定的问题，这里可以把 dr^2 作为差异度信息熵计算的权重值引入公式，之所以选择平方，是由于 dr 存在负值。同时考虑用户共同评分项目集合 $SaRU(u_i,u_j)$，用户共同评分集合越大，用户间相似性越高，否则越低，因此考虑把 L 作为差异度信息熵计算的权重值引入公式对其进行优化，因此，调整后的加权差异度信息熵 $URDE(u_i,u_j)$ 计算公式如4.7所示。

$$URDE(u_i,u_j)=\frac{1}{L}\sum_{i=1}^{n}p(dr_i)log_2\frac{1}{p(dr_i)}\times dr_i^2 \tag{4.7}$$

对于用户 u_i，分别计算 $URDE(u_i,u_j), j\in[1,m], j\neq i$ 得到用户 u_i 的加权差异度信息熵向量 $\mathbf{URDE}_i$。

Step 4：归一化用户加权差异度信息熵

通过公式4.7计算得到的用户加权差异度信息熵，其结果范围不在正常相似性范围内，因此需要对 $\mathbf{URDE}_i$ 进行归一化处理。归一化处理的方式很多，如高斯函数、*tanh*函数、*Sigmoid*函数等等，本文采用的是极值线性模型对其进行归一化的处理。具体如公式4.8所示。

$$\mathbf{URDE}_i[k]=\frac{Max(URDE(u_i,u_j))-URDE(u_i,u_k)}{Max(URDE(u_i,u_j))-Min(URDE(u_i,u_k))}, j\in[1,m], j\neq i \tag{4.8}$$

$Max(URDE(u_i,u_j))$ 表示用户 u_i 与其它用户间加权差异度信息熵的最大值，$Min(URDE(u_i,u_j))$ 表示用户 u_i 与其它用户间加权差异度信息熵的最小值。$WDE(u_i,u_j)$ 中的元素最小值，$\mathbf{URDE}_i[k]$ 表示用户 u_i 与用户 u_k 归一化后的加权差异度信息熵。

针对上面的算法过程，具体的算法实现如表4.1。

表4.1 基于用户评分差异度信息熵的相似性计算算法

Table. 4.1 The algorithm of similarity calculation algorithm based on user rating difference information entropy

Algorithm 4.1 基于用户评分差异度信息熵的相似性计算算法

Input：用户评分矩阵 $R_{m\times n}$

Output：基于用户评分差异度信息熵的相似性矩阵 $URDE_{m\times m}$

Step:

1：*Begin*

2：　$r_i\leftarrow R_{m\times n}, r_j\leftarrow R_{m\times n}$ //抽取用户 i 和用户 j 的评分

3：　$SaRU(u_i,u_j)\leftarrow r_i\cap r_j$ //用户 i 和 j 的共同评分项目

4:　*for(k in SARU)*

5:　　$r_{i,k}\leftarrow r_{i,k}\text{-}\bar{r}_i+\delta$ //消除用户自身评分偏好的影响

续表4.1 基于用户评分差异度信息熵的相似性计算算法

Continued Table. 4.1 The algorithm of similarity calculation algorithm based on user rating difference information entropy

6:	$dr_k \leftarrow r_{i,k}\text{-}r_{j,k}$ //用户评分差异度
7:	*end for*
8:	*for*(计算 *DR* 中各级别评分数据的概率 $p(dr_i)$)
9:	$DR(u_i,u_j) \leftarrow dr_k +$ //差异度集合
10:	$p(dr_i) \leftarrow fre(dr_i)/fre$
11:	*end for*
12:	*for*(*j in 1:m*)//计算加权差异度信息熵
13:	*for*(*j in 1:m*)
14:	$URDE(u_i,u_j) \leftarrow -1/L * p(dr_i) * log_2 p(dr_i) * pow(2,dr)$
15:	*end for*
16:	*end for*
17:	*for*(*i in 1:m*)//加权差异度信息熵归一化
18:	*for*(*j in 1:m*)
19:	$m1 \leftarrow Max(URDE(u_i,u_j)) - URDE(u_i,u_k)$
20:	$m2 = Max(URDE(u_i,u_j)) - Min(URDE(u_i,u_k))$
21:	$URDE_i[j] \leftarrow m1/m2$
22:	*end for*
23:	*end for*
24：	*return URDE*
25:	*End*

4.3.2 基于用户项目兴趣度的相似性计算方法

（1）基于用户项目兴趣度的相似性计算方法基本思想

传统的协同过滤推荐结果往往存在着推荐准确率、召回率不高的情况，这主要是由于基于用户的协同过滤推荐算法以及基于项目的协同过滤算法中相似性计算只考虑到用户之间、项目之间的关系，并没有考虑到用户具体的行为对用户兴趣的影响，同时没有考虑到对推荐结果的反馈优化。本文在传统协同过滤的基础上考虑了这些问题并应用到相似性计算过程中，优化最终的推荐质量。

用户对推荐结果的直接交互行为往往间接地反映了用户的兴趣信息，用户对推荐Item的鼠标操作行为，包括观看行为等都在一定程度上反映了用户的偏好信息。因此，

给出兴趣度的定义，如定义4.1。用于衡量用户对具体推荐项目的兴趣程度，兴趣度越大，表明用户对具体项目的兴趣程度越高，兴趣度越小，表明用户对具体项目的兴趣程度越低。通过用户兴趣度的计算形成用户兴趣度矩阵。这里提出基于用户项目兴趣度的相似性计算方法用于计算两个用户兴趣度之间的相似性。进一步优化上一小节中提出的基于差异度信息熵的相似性计算方法。

（2）基于用户项目兴趣度的相似性计算方法详细过程

基于用户项目兴趣度的相似性计算方法主要是通过用户之间兴趣来计算相似性，能更好地表达用户之间相似程度。具体的实现过程如图4.3所示。

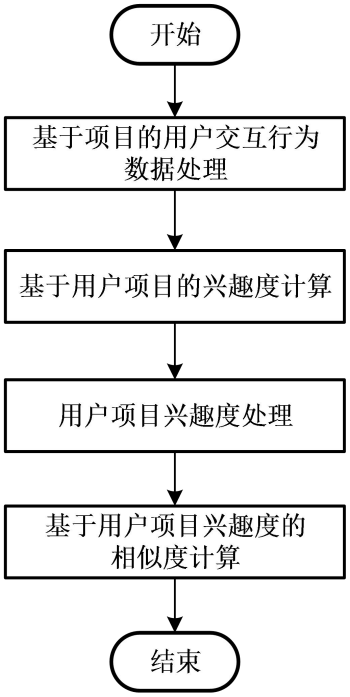

图 4.3 基于用户项目兴趣度的相似性计算方法过程图

Fig. 4.3 The process of similarity calculation method based on user item interest degree

此相似性计算方法通过三个步骤完成：

Step 1：基于用户项目的兴趣度矩阵

基于用户项目的兴趣度计算方法将在第五章5.3.3中给出详细的过程，这里给出得到的用户兴趣度矩阵如公式4.9所示。其中，IDI_{ij} 表示用户 u_i 对类标签 $Item_j$ 的兴趣度。

$$IDI = \begin{array}{c} \\ u_1 \\ u_2 \\ \cdots \\ u_i \\ \cdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} Item_1 & Item_2 & \cdots & Item_n \end{array} \\ \begin{bmatrix} IDI_{11} & IDI_{12} & \cdots & IDI_{1n} \\ IDI_{21} & IDI_{22} & \cdots & IDI_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ IDI_{i1} & IDI_{i2} & \cdots & IDI_{in} \\ \cdots & \cdots & \cdots & \cdots \\ IDI_{m1} & IDI_{m2} & \cdots & IDI_{mn} \end{bmatrix} \end{array} \tag{4.9}$$

这里定义用户集合 $U=\{u_1,u_2,...,u_m\}$，推荐列表集合 $Item=\{Item_1,Item_2,...,Item_n\}$，其中矩阵的 IDI_{ij} 代表用户 u_i 对推荐列表 $Item_j$ 的兴趣度。

Step 2：基于用户项目的兴趣度向量的生成

这里定义基于用户项目的兴趣度向量 $\overrightarrow{IDI_i}$ 如公式4.10所示

$$\overrightarrow{IDI_i}=\begin{matrix} & Item_i & ... & Item_j & ... & Item_m \\ u_i & [IDI_{i1} & ... & IDI_{ij} & ... & IDI_{im}] \end{matrix} \tag{4.10}$$

定义用户平均兴趣度 $\overline{IDI_i}$ 如4.11所示

$$\overline{IDI_i}=\frac{1}{m}\sum_{j=1}^{m}IDI_{ij} \tag{4.11}$$

Step 3：基于用户项目兴趣度的相似性计算

设用户 u_i 和 u_j 的共同推荐Item的集合用 $Item_{ij}$ 进行表示，那么通过皮尔森（Pearson）相关系数对用户 u_i 和 u_j 进行基于用户项目兴趣度的相似性 $ITSim(u_i,u_j)$ 计算公式如4.12所示。

$$ITSim(u_i,u_j)=\frac{\sum_{c\in Item_{ij}}(IDI_{ic}-\overline{IDI_i})(IDI_{jc}-\overline{IDI_j})}{\sqrt{\sum_{c\in Item_{ij}}(IDI_{ic}-\overline{IDI_i})^2}\sqrt{\sum_{c\in Item_{ij}}(IDI_{jc}-\overline{IDI_j})^2}} \tag{4.12}$$

4.3.3 基于用户类别兴趣度的相似性计算方法

（1）基于用户类标签的相似性计算方法基本思想

传统的协同过滤推荐结果往往存在着推荐准确率、召回率不高的情况，这主要是由于基于用户的协同过滤推荐算法以及基于项目的协同过滤算法中相似性计算只考虑到用户之间、项目之间的关系，并没有考虑到用户具体的行为对用户兴趣的影响，同时没有考虑到对推荐结果的反馈优化。本文考虑在交互的过程中让用户的主观交互式行为参与进来，并应用到相似性计算过程中，优化最终的推荐质量。

本文通过设置推荐项目的类标签，让用户通过鼠标拖动距离这些类标签的远近表达个人更感兴趣的类别，提出基于用户类标签的相似性计算方法，不同于4.3.2章节中的方法，这里主要是通过用户主观动作直观表达用户兴趣，通过计算用户的兴趣度衡量用户针对类标签的兴趣程度。

兴趣度越大，表明用户对类标签Lable的兴趣程度越高。兴趣度越小，表明用户对类标签Lable的兴趣程度越低。通过用户兴趣度的计算形成用户兴趣度矩阵。这里提出基于用户类标签的相似性计算方法用于计算两个用户兴趣度之间的相似性。进一步优化4.3.1章节中提出的基于用户评分差异度信息熵的相似性计算方法。

（2）基于用户类标签的相似性计算方法详细过程

基于用户类标签的相似性计算方法主要是通过用户之间兴趣来计算相似性，能更好地表达用户之间的相似程度。具体的实现过程如图4.4所示。

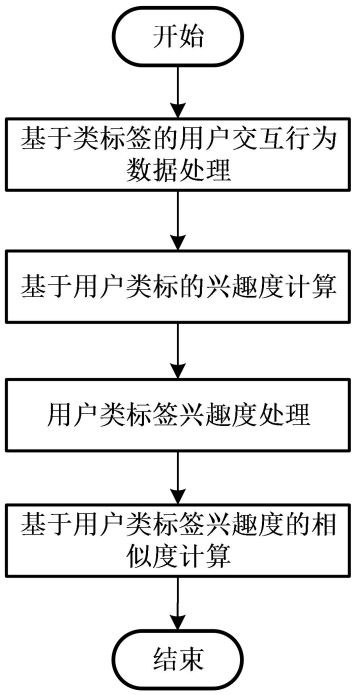

图 4.4 基于用户类标签兴趣度的相似性计算方法过程图

Fig. 4.4 The process of similarity calculation method based on user category label interest degree

此相似性计算方法通过三个步骤完成：

Step 1：基于用户类标签的兴趣度矩阵

具体的基于类标签的兴趣度计算方法将在第五章5.3.4中给出详细的过程，这里给出通过计算得到的基于用户类标签兴趣度矩阵如公式4.13所示。其中，IDL_{ij} 表示用户 u_i 对类标签 Lab_j 的兴趣度。

$$IDL = \begin{array}{c} \\ u_1 \\ u_2 \\ \cdots \\ u_i \\ \cdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} Lab_1 & Lab_2 & \cdots & Lab_k \end{array} \\ \begin{bmatrix} IDL_{11} & IDL_{12} & \cdots & IDL_{1k} \\ ID_{21} & ID_{22} & \cdots & IDL_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ IDL_{i1} & IDL_{i2} & \cdots & IDL_{ik} \\ \cdots & \cdots & \cdots & \cdots \\ IDL_{m1} & IDL_{m2} & \cdots & IDL_{mk} \end{bmatrix} \end{array} \tag{4.13}$$

这里定义用户集合 $U=\{u_1,u_2,...,u_m\}$，类别标签集合 $Lab=\{Lab_1,Lab_2,...,Lab_k\}$，其中矩阵的 IDL_{ij} 代表用户 u_i 对类别标签 Lab_j 的兴趣度。

Step 2：基于用户类标签的兴趣度向量的生成

定义用户平均兴趣度 $\overline{IDL_i}$ 如4.14所示。

$$\overline{IDL_i} = \frac{1}{m}\sum_{j=1}^{m} IDL_{ij} \tag{4.14}$$

定义基于用户类标签的兴趣度向量 $\overrightarrow{IDL_i}$ 如公式4.15所示.

$$\overrightarrow{IDL_i} = \begin{matrix} \\ u_i \end{matrix} \begin{matrix} Lab_i & \dots & Lab_j & \dots & Lab_k \\ [IDL_{i1} & \dots & IDL_{ij} & \dots & IDL_{ik}] \end{matrix} \tag{4.15}$$

Step 3：基于用户类标签兴趣度的相似性计算

设用户 u_i 和 u_j 的共同感兴趣标签的集合用 Lab_{ij} 进行表示，那么通过皮尔森（Pearson）相关系数对用户 u_i 和 u_j 进行基于用户类别标签兴趣度的用户兴趣相似性 $ILSim(u_i,u_j)$ 公式如4.16所示。

$$ILSim(u_i,u_j) = \frac{\sum_{c\in Lab_{ij}}(ID_{ic}-\overline{ID_i})(ID_{jc}-\overline{ID_j})}{\sqrt{\sum_{c\in Lab_{ij}}(ID_{ic}-\overline{ID_i})^2}\sqrt{\sum_{c\in Lab_{ij}}(ID_{jc}-\overline{ID_j})^2}} \tag{4.16}$$

4.3.4 基于用户交互行为兴趣度的相似性计算方法

通过 4.3.2 章节中的基于用户项目兴趣度的相似性计算方法求出 $ItSim(u_i,u_j)$，4.3.3 章节中的基于用户类别标签兴趣度的相似性计算方法求出 $ILSim(u_i,u_j)$，为了更好的对用户相似性进行衡量，本文提出了基于用户交互行为兴趣度的相似性计算方法，如定义 4.3 所示。

【定义4.3】基于用户交互行为兴趣度的相似性计算方法(UIBD)(A Similarity Calculation Method based on the User Interaction Behavior Interest Degree)：本文将基于UIBD的相似性计算方法形式化定义为一个二元组*UIBDSim*=<*ITSim*,*ILSim*>，定义用户 u_i 与用户 u_j 之间的相似性 $UIBDSim(u_i,u_j)$ 如公式4.17所示。

$$UIBDSim(u_i,u_j) \leftarrow \omega_1 \times ITSim(u_i,u_j) + \omega_2 \times ILSim(u_i,u_j) \tag{4.17}$$

其中 ω，作为权重值 $(0 \le \omega_i \le 1, i=1,2)$，且 $(\sum \omega_i = 1, i=1,2)$ 用于调整基于两种不同用户交互行为相似性计算方法之间的关系，通过初始推荐之后获取用户交互式行为兴趣度，如果用户的初始用户评分数据不是很多，此时 ω_2 的权重可以适当调整较大来进行优化推荐。

针对上面的算法过程，具体的算法如表4.2所示。

表 4.2 基于用户交互行为兴趣度的相似性计算方法

Table. 4.2 The algorithm of similarity calculation based on user interaction behavior interest degree

Algorithm 4.2 基于用户交互行为兴趣度的相似性计算方法

Input：基于用户项目的兴趣度矩阵 $IDI_{m\times n}$，基于用户类标签的兴趣度矩阵 $IDL_{m\times k}$

续表 4.2 基于用户交互行为兴趣度的相似性计算方法

Continued Table. 4.2 The algorithm of similarity calculation based on user interaction behavior interest degree

Algorithm 4.2 基于用户交互行为兴趣度的相似性计算方法

Output：基于用户交互行为的相似性矩阵 $UIBDSim_{m\times m}$

Step:

1：*Begin*

2：　*for(i in 1:m)* //分别计算每个用户的平均兴趣度

3:　　*for(j in 1:n)*

4:　　$\overline{IDI_i} \leftarrow 1/m * IDI_{ij} +$ //分别计算每个用户基于项目的平均兴趣度

5:　　*end　for*

6:　　*for (i in 1:k)*

7:　　$\overline{IDL_i} \leftarrow 1/k * IDL_{ij} +$ //分别计算每个用户基于类标签的平均兴趣度

8:　　*end　for*

9:　*end　for*

10:　*for(i in 1:m)* //分别计算用户间项目兴趣度相似性以及类标签兴趣度相似性

11:　　*for(j in 1:m)*

12:　　$ItSim(u_i,u_j) \leftarrow Pearson(IDI,\overline{IDI})$

13:　　$LabSim(u_i,u_j) \leftarrow Pearson(IDL,\overline{IDL})$

14:　　$UIBSim(u_i,u_j) \leftarrow \omega_1 * ITSim(u_i,u_j) + \omega_2 * ILSim(u_i,u_j)$

15:　　*end　for*

16:　*end　for*

17:　*return UIBDSim*

18: *End*

4.4 本章小结

本章首先对目前协同过滤中存在的数据稀疏性问题进行了分析，在此基础上提出基于用户评分差异度信息熵的相似性计算方法、基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，并分别对每个方法进行详细的说明，其中，通过计算用户项目兴趣度的相似性以及用户类标签兴趣度的相似性，用于基于用户交互行为兴趣度的相似性方法中。

第 5 章 基于推荐结果的用户交互行为兴趣度计算方法

用户评分是用户对物品喜好的直观反映，协同过滤算法就是把用户评分数据作为推荐的依据。但是这种方法忽略了用户行为数据中的隐含信息以及用户偏好随时间变化等特征。传统推荐技术存在两方面的不足：一方面缺少对于用户行为的重视，仅把用户评分作为推荐依据；另一方面，对于用户的兴趣随着时间的推移会发生改变这一因素考虑不足。

本章对基于推荐结果的用户交互行为兴趣度进行了详细分析。本章主要将用户交互行为分为两类，计算得到其对应兴趣度，形成用户兴趣度矩阵。考虑用户兴趣随时间变化的情况，使用用户兴趣调整函数进行更新。

5.1 相关工作

为了更好地使用协同过滤对用户进行推荐，不仅需要了解其对哪些物品感兴趣，而且还需要了解其兴趣度。传统获取用户感兴趣内容的方式主要是通过用户的参与，即用户对自己感兴趣的内容主动进行相关标注，比如一些系统在用户初始进入时，会要求用户输入自己感兴趣的标签，以此获取用户初始兴趣，进行相应推荐。这种方式虽然直观，但并不是每个用户都能准确描述个人的兴趣，所以获取的兴趣也会出现一定偏差，同时，由于这种方式会对用户的体验产生一定影响，往往用户的参与度并不高，获取的数据并不多，因此，这种方式并不是特别好。另一用获取用户感兴趣内容的方式是通过用户与系统的交互来分析用户的潜在兴趣，这种方式往往不会影响用户的体验，可以提高系统的友好程度。

在心理学中，认为一个人的行为可以体现其兴趣。同样，针对互联网来说，用户的网络行为也可以反映用户的兴趣。例如，用户在浏览淘宝商品时，用户对其感兴趣的商品进行点击、收藏、加入购物车等行为，这些行为可以反映用户对其感兴趣；同样，如果用户频繁访问某个商品，或者对其反复浏览，停留时间比较长都可以认为对其比较感兴趣。在文献[37] [38] [39]中，对用户的各种行为隐含的兴趣信息进行了研究。

本文在上述工作的基础上，提出本文的用户交互行为，具体的定义在第三章已经给出，主要是指 UAIB、UWMB、 UOMB、UMDB，通过这些交互行为来对用户兴趣度进行衡量。

5.2 基于推荐结果的用户交互行为兴趣度计算基本思路

本节主要是对本章提出的基于推荐结果的用户交互行为兴趣度计算基本处理过程进行描述。

5.2.1 问题描述

目前的协同过滤推荐大都是一种静态的推荐，即根据用户的评分矩阵进行计算以推荐给用户新的可能感兴趣的物品，但是，推荐是一个交互式的动态过程。大量研究表明用户的实际兴趣与交互行为是密切相关的，通过对用户交互行为分析可以获取用户兴趣信息，通过计算用户兴趣度，反馈到协同过滤过程中从而使推荐结果更加贴近用户的期望。

基于此，本章提出基于推荐结果的用户交互行为兴趣度计算方法，该方法基于URDE方法得到的推荐结果，通过用户交互行为表达个人兴趣。考虑用户对推荐列表不同交互行为，本章提出基于用户项目的兴趣度计算方法以及基于用户类标签的兴趣度计算方法分别用于计算用户对项目的感兴趣程度和用户对类标签的感兴趣程度。在此基础上生成用户兴趣度矩阵，解决用户的推荐反馈问题。

5.2.2 问题解决的基本思路

本章提出基于推荐结果的用户交互行为兴趣度计算方法的解决思路如图 5.1 所示。根据第四章 URDE 的方法生成的推荐列表，用户对其进行交互，获取用户的交互行为，本章的用户交互行为主要是针对两种：基于用户项目的交互行为以及基于用户类标签的交互行为，分别对应两种不同的兴趣度计算方法，通过对其计算兴趣度得到兴趣度矩阵，最后通过用户兴趣度调整函数对用户兴趣度进行调整，形成用户兴趣度矩阵。将用户的推荐反馈问题转化成用户兴趣度的计算。

这里包含两个主要问题：

（1） 如何对用户不同交互行为计算兴趣度

用户兴趣度是对用户兴趣的一个直接度量，找到合适的方法对用户的兴趣度进行计算至关重要。考虑用户交互行为的对象以及方式，本文将用户的交互行为分为基于用户的交互行为以及基于用户类标签的交互行为，通过不同的方法进行兴趣的计算。基于用户类标签的兴趣度计算方法主要是考虑用户的主观参与，通过主观行为，来进行兴趣度的计算。基于项目的兴趣度计算主要是考虑用户的间接参与，用户通过对其推荐的项目的各种操作来间接反映其兴趣度。最后形成用户兴趣度矩阵。

（2） 如何对用户兴趣度更新

用户的个人兴趣随着时间的变化会发生改变，这就是“兴趣漂移”，这是一个自然现象，艾比豪斯就对此进行了相关研究。同样本文在得到用户兴趣度矩阵之后，需要考虑本文的用户兴趣度影响因素，这里主要考了时间的因素，对其进行调整，从而优化兴趣度矩阵。

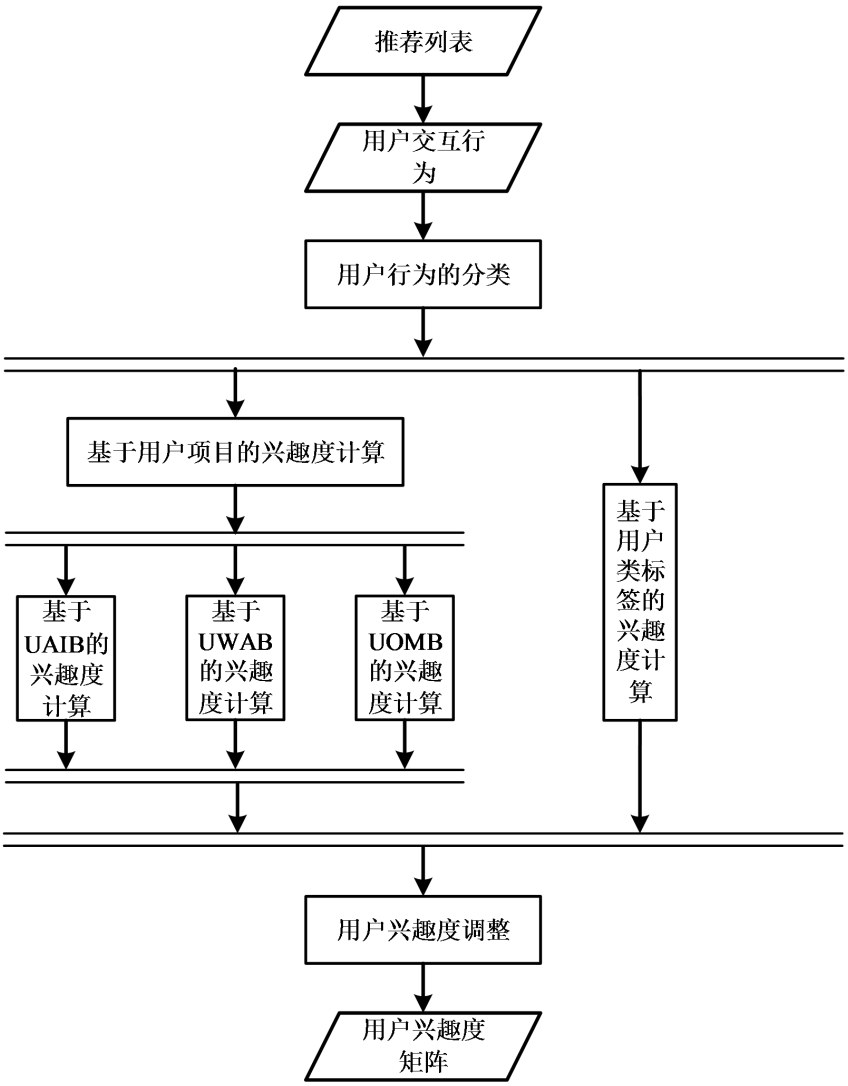

图 5.1 基于推荐结果的户交互行为兴趣度计算方法过程图

Fig. 5.1 The process of Interest Degree Computing Method of User Interaction based on recommended result

5.3 基于推荐结果的用户交互行为兴趣度计算方法

为了对计算基于推荐结果的用户交互行为兴趣度，本文给出了用户行为序列的定义，具体如定义5.1所示。通过用户之间行为序列可以计算用户对项目的感兴趣程度，进而通过引入协同过滤算法发现用户近邻，进行相关项目的推荐，这里的用户行为主要是用户

的交互行为，一方面用户的交互行为是对个人兴趣的直接调整，另一方面用户的交互行为是对用户兴趣的间接反映。这里通过兴趣度衡量用户的感兴趣程度。

【定义 5.1】用户行为序列（User Behavior Sequence）：对于用户 u_i，其对某一项目一段时间的行为可以表示为一个行为序列，本文对其形式化定义为：$BS=<U,Item,MID,<BeID_1,fre>,<BeID_2,fre>,...,<BeID_n,fre>,ST,ET>$，其中：

- $Item=\{Item_1,Item_2,...,Item_m\}$ *是推荐列表条目的集合；*
- $U=\{u_1,u_2,...,u_m\}$ *是用户集合；*
- *MID* 为推荐列表条目对应项目页面 *Page*；
- $BeID_1,BeID_1,...,BeID_n$ 为用户不同交互式行为的 ID；
- *fre* 为交互行为对应的次数
- ST 为行为序列开始时间；
- ET 为行为序列结束时间；

用户不同的交互行为对应不同的用户兴趣度矩阵，这里通过用户交互行为兴趣度模型来进行描述，具体如定义5.2所示。

【定义5.2】用户交互行为兴趣度模型 (User Interactive Behavior Interest Model)：通过获取用户与推荐列表的交互式行为来进行用户兴趣计算，针对具体的兴趣度计算方法，本章提出用户交互行为兴趣度模型的概念。该模型形式化定义为一个二元组 $UIIM=<IDI,IDL>$，将兴趣度计算分为两个部分，其中：

- IDI 代表基于用户项目的兴趣度。
- IDL 代表基于用户类标签的兴趣度。

下面将对这两种兴趣度进行详细说明。

5.3.1 基于用户项目的兴趣度计算方法

（1）基于 UAIB 的兴趣度计算

在本文中，通过 URDE 方法最终的出的推荐列表，用户会对其感兴趣的推荐列表条目 Item 进行访问操作，这也间接隐含了用户的兴趣信息，因此通过用户对推荐列表的访问行为去计算其兴趣度是可行的，本文通过定义 3.4 对用户对项目的访问行为 UAIB 进行定义，其中，$Fre(a)$ 表示用户对某一推荐列表的访问次数。

$$AcID_{fre}(a)\propto Fre(a) \tag{5.1}$$

其中，$AcID_{fre}(a)$ 表示用户基于 UAIB 的兴趣度，这里定义其计算公式如 5.2 所示。

$$AcID_{fre}(a)=\frac{Fre(a)}{\max\limits_{a\in A} Fre(a)} \tag{5.2}$$

公式 5.2 只是一种定量的兴趣度计算方法，随着时间的累积，UAIB 次数会逐渐增加，但是这些累积的访问次数由于时间的远近并不能很好地反映用户当前的兴趣，因此需要对用户访问的时间进行限制，在本文中设置一周为一个更新周期，进行 UAIB 次数的更新。

具体的更新方法如公式 5.3 所示。

$$p=(|Fre_{new}(a)-Fre_{old}(a)|)/Fre_{old}(a) \tag{5.3}$$

公式 5.3 中，$Fre_{old}(a)$ 是用户对推荐列表的上一周期的访问次数，$Fre_{new}(a)$ 则是目前的统计次数，p 推荐列表条目访问次数的更新比例。

若 $p<0.5$，即用户在上一个统计周期和当前统计周期中对某一推荐列表条目点击次数的变化程度不足，则认为用户对该推荐列表条目的兴趣随着时间的推移并未有较大的转移，如公式 5.4 所示。

$$Fre(a)=(|Fre_{new}(a)-Fre_{old}(a)|)/2 \tag{5.4}$$

若 $p\geq 0.5$，即用户在上一个统计周期和当前统计周期中对某一推荐列表条目点击变化程度较大，则认为用户对该推荐列表条目访问的兴趣已经随着时间的推移有了较大的改变，如公式 5.5 所示。

$$Fre(a)=Fre_{new}(a) \tag{5.5}$$

由于用户兴趣度是用户对项目兴趣程度的一种度量，这里规定用户兴趣度在[0,1]之间，通常情况下，UAIB 次数的差别较大，如果仅是简单的线性归一化处理，容易造成基于 UWAB 的兴趣度接近于 0 而远小于 1，因此，本文进一步对其进行归一化处理，如公式 5.6 所示。

$$AcID_{fre}(a)=\frac{log_2(Fre(a))-\min\limits_{w\in W}\{log_2(Fre(a))\}}{\max\limits_{w\in W}\{log_2(Fre(a))\}-\min\limits_{w\in W}\{log_2(Fre(a))\}} \tag{5.6}$$

经过公示 5.6 的处理，$0\leq AcIF_{fre}(a)\leq 1$

（2）基于 UWAB 的兴趣度计算

用户会对其感兴趣的影片进行观看，因此通过用户的观看影片行为去计算用户的兴趣度是可行的，本文通过定义 3.5 对用户对项目的影片观看行为 UWMB 进行定义。对于用户观看某一感兴趣影片信息而言，用户观看时间越长，表明用户对该推荐列表条目

越感兴趣，但由于不同影片的时长不同，这里引入用户对项目的影片观看时长占比的概念对其进行更好的衡量，观看时长占比的具体定义如定义 5.3 所示

【定义 5.3】用户对项目的影片观看时长占比（the Proportion of User Watching Movies on the Item）：对于影片时长为$Time_{movie}$，用户观看时长为$Time_{watch}$，其观看时长占比定义如公式 5.7 所示。

$$TimePro(w) = Time_{watch} / Time_{movie} \tag{5.7}$$

对于$WaID_{CFTimePro}(w)$而言，对于推荐列表条目 Item 所对应的 Page，$TimePro(w)$越大，用户兴趣度$WaID_{CFTimePro}(w)$就越大。如公式 5.8 所示。

$$WaID_{TimePro}(w) \propto TimePro(w) \tag{5.8}$$

本文定义基于 UWAB 的兴趣度计算如公式 5.9 所示。

$$WaID_{TimePro}(w) = \frac{TimePro(w)}{\max\limits_{w \in W} TimePro(w)} \tag{5.9}$$

由于用户兴趣度是用户对项目兴趣程度的一种度量，这里规定用户兴趣度在[0,1]之间，通常情况下，UWAB 的差别较大，如果仅是简单的线性归一化处理，容易造成基于 UWAB 的兴趣度接近于 0 而远小于 1，因此，本文进一步对其进行归一化处理，如公式 5.10 所示。

$$WaID_{TimePro}(w) = \frac{log_2(TimePro(w)) - \min\limits_{w \in W}\{log_2(TimePro(w))\}}{\max\limits_{w \in W}\{log_2(TimePro(w))\} - \min\limits_{w \in W}\{log_2(TimePro(w))\}} \tag{5.10}$$

经过公示 5.10 的处理，$0 \le WaID_{TimePro}(w) \le 1$

（3）基于 UOMB 的兴趣度计算

用户对其感兴趣的信息会进行一系列鼠标操作，因此通过用户的鼠标操作去计算用户的兴趣度是可行的，但仅是鼠标操作并不能对用户兴趣度进行很好地度量，本文通过定义 3.6 对用户对项目的鼠标操作行为 UOMB 进行了定义。

这里通过 *Fre(ml)*记录鼠标左键的点击次数，通过 *Fre(mr)*记录鼠标的右键操作次数，通过 *Fre(pd)*记录按下 PageDown 的次数，通过 *Fre(pul)*记录按下 PageUp 的次数，通过 *Fre(cc)*记录光标键产生的滚动次数，通过 *Fre(cl)*记录鼠标滚轮产生的滚动次数

定义用户鼠标操作次数的计算方法如公式 5.11 所示。

$$Fre(m) = Fre(ml) + Fre(mr) + Fre(pd) + Fre(pu) + Fre(cc) + Fre(mc) \tag{5.11}$$

对于推荐列表条目 Item，$Fre(m)$越大，用户兴趣度就越大。如公式 5.12 所示。

$$MOID_{MO}(m) \propto Fre(m) \tag{5.12}$$

本文定义基于 UOMB 的兴趣度计算如公式 5.13 所示。

$$MOID_{MO}(m) = \frac{Fre(m)}{\max\limits_{m \in M} Fre(m)} \tag{5.13}$$

由于用户兴趣度是用户对项目兴趣程度的一种度量，这里规定用户兴趣度在[0,1]之间，通常情况下，UOMB 的差别较大，如果仅是简单的线性归一化处理，容易造成基于 UOMB 的兴趣度接近于 0 而远小于 1，因此，本文进一步对其进行归一化处理，如公式 5.14 所示。

$$MOID_{MO}(m) = \frac{log_2(Fre(m)) - \min\limits_{m \in M}\{log_2(Fre(m))\}}{\max\limits_{m \in M}\{log_2(Fre(m))\} - \min\limits_{m \in M}\{log_2(Fre(m))\}} \tag{5.14}$$

经过公示 5.14 的处理，$0 \le MOID_{MO}(m) \le 1$

（4）用户最小交互行为组合兴趣度

这里提出用户最小交互行为组合兴趣度的方法来对用户的上述三种交互行为的兴趣度进行衡量，对于一个用户来说，用户不同的交互行为体现用户不同的兴趣，这里设 λ，η，β 分别为基于 *UAIB, UWAB* 以及 *UOMB* 三种交互行为兴趣度的权重，可以调整权重提高兴趣度计算的精度。定义基于用户项目的兴趣计算函数 $IDI(u_i, Item_j)$ 计算方法如公式 5.15 所示。

$$IDI(u_i, Item_j) = \lambda \cdot AcIF_{fre}(a) + \eta \cdot WaID_{TimePro}(w) + \beta \cdot MOID_{MO}(m) \tag{5.15}$$

其中 $\lambda + \eta + \beta = 1, 0 \le (\lambda, \eta, \beta) \le 1$，通过公式 5.15 的计算形成基于用户项目的兴趣度矩阵如公式 5.16 所示。

$$IDI = \begin{array}{c} \\ u_1 \\ u_2 \\ \cdots \\ u_i \\ \cdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} Item_1 & Item_2 & \cdots & Item_n \end{array} \\ \begin{bmatrix} IDI_{11} & IDI_{12} & \cdots & IDI_{1n} \\ IDI_{21} & IDI_{22} & \cdots & IDI_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ IDI_{i1} & IDI_{i2} & \cdots & IDI_{in} \\ \cdots & \cdots & \cdots & \cdots \\ IDI_{m1} & IDI_{m2} & \cdots & IDI_{mn} \end{bmatrix} \end{array} \tag{5.16}$$

此用户兴趣度矩阵用户 4.3.2 节中基于用户项目兴趣度的相似性计算中。

具体的基于用户项目的兴趣度计算方法如表 5.1 所示。

表5.1 基于用户项目的兴趣度计算方法

Table. 5.1 The calculation method of interest degree based on user

Algorithm 5.1 基于用户项目的兴趣度计算方法

Input：用户交互行为矩阵 $UOMB_{m\times n}, UWAB_{m\times n}, UAIB_{m\times n}$

Output：用户兴趣度矩阵 $IDI_{m\times n}$

Step:

1：*Begin*

2：　*for(i in 1:m)*

3：　　*for(j in 1:n)*

4:　　分别计算用户对项目的 $AcID_{i,j}(a)$、$WaID_{i,j}(w)$、$MOID_{i,j}(m)$

5:　　进行归一户处理

6:　　*end　for*

7:　*end　for*

8:　*for*(*i in 1:m*)//计算用户最小交互行为组合兴趣度

9:　　*for(j in 1:n)*

10:　　$IDI_{i,j} \leftarrow= \lambda \cdot AcIF_{i,j}(a) + \eta \cdot WaID_{i,j}(w) + \beta \cdot MOID_{i,j}(m)$

11:　　*end　for*

12:　*end　for*

13:　*return IDI*

14: *End*

5.3.2 基于用户类标签的兴趣度计算方法

本文通过定义 3.3 对用户对类标签的鼠标拖动行为 UMDB 进行定义，用于衡量用户类标签兴趣度。对于用户 u_i，用户通过拖动鼠标使距其感兴趣的类标签距离更近，表明用户对该类标签兴趣更大。

如图 5.2 所示，其中，类标签集合 $Lab=\{L_1, L_2, ..., L_k\}$，用户 u_i 通过拖动鼠标改变其兴趣标签，拖动后为 $u_i^{'}$，设用户初始的位置坐标为（x_{u_i}，y_{u_i}），拖动后的位置坐标为（$x_{u_i}^{'}$，$y_{u_i}^{'}$），标签 L_i 的位置坐标为（x_{Lj}，y_{Lj}）。

根据欧式距离公式，则拖动之前的欧氏距离如公式 5.17 所示。

$$DisA=\sqrt{(x_{u_i}-x_j)^2+(y_{u_i}-y_j)^2} \tag{5.17}$$

拖动之后的欧氏距离如公式 5.18 所示。

$$DisB = \sqrt{(x'_{u_i} - x_j)^2 + (y'_{u_i} - y_j)^2} \tag{5.18}$$

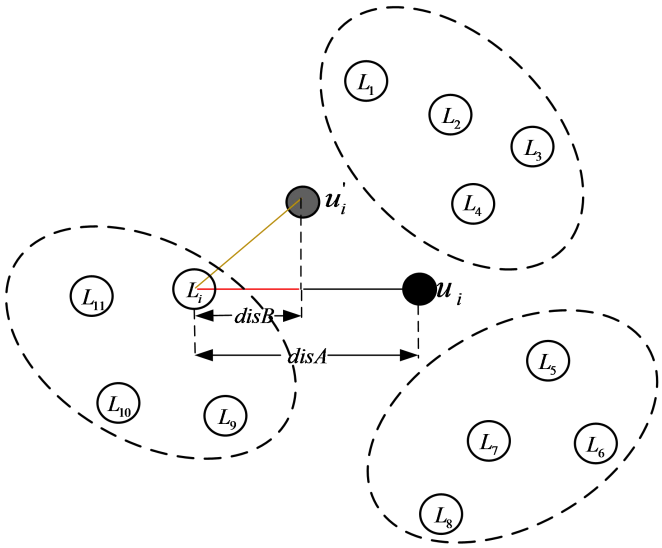

图5.2 基于用户类标签的兴趣度计算方法示意图

Fig. 5.2 Interest degree computing method based on user category labels

但由于用户和不同类标签的初始距离不同，移动距离也不相同，因此这里引入用户拖动距离占比的概念，具体如定义 5.4 所示。

【定义 5.4】用户拖动距离占比（the User Drag Distance Proportion）： 用户 u_i 距离类标签 Lab_j 的初始距离为 $DisA$，通过拖动 u_i 距离类标签 Lab_j 的距离为 $DisB$，则本文定义距离占比如公式 5.19 所示。

$$DisPro(l) = \frac{DisB(u'_i, Lab_j)}{DisA(u_i, Lab_j)} \tag{5.19}$$

对类别标签 Lab_j 而言， $DisPro(l)$ 越大，基于用户标签的兴趣度 $IDL(u_i, Lab_j)$ 就越大。如公式 5.20 所示。

$$IDL(u_i, Lab_j) \propto DisPro(l) \tag{5.20}$$

因此这里通过公式 5.21 计算用户对类标签的兴趣度。

$$IDL(u_i, Lab_j) = \frac{DisPro(l)}{\max\limits_{l \in L} DisPro(l)} \tag{5.21}$$

通常情况下，用户鼠标拖动行为(UMDB)，如果仅是简单的线性归一化处理，容易造成基于 UMDB 的兴趣度接近于 0 而远小于 1，因此，本文进一步对其进行归一化处理，如公式 5.22 所示。

$$IDL(u_i, Lab_j) = \frac{log_2(IDL(u_i, Lab_j)) - \min\limits_{j\in[1,k]}\{log_2(IDL(u_i, Lab_j))\}}{\max\limits_{j\in[1,k]}\{log_2(IDL(u_i, Lab_j))\} - \min\limits_{j\in[1,k]}\{log_2(IDL(u_i, Lab_j))\}} \tag{5.22}$$

通过归一化处理之后，$0 \le IDL(u_i, Lab_j) \le 1$，得到用户对类标签的兴趣度。

通过公式 5.22 的计算形成基于标签的用户兴趣度矩阵如公式 5.23 所示。

$$IDL = \begin{array}{c} \\ u_1 \\ u_2 \\ \cdots \\ u_i \\ \cdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} Lab_1 & Lab_2 & \cdots & Lab_k \end{array} \\ \begin{bmatrix} IDL_{11} & IDL_{12} & \cdots & IDL_{1k} \\ ID_{21} & ID_{22} & \cdots & IDL_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ IDL_{i1} & IDL_{i2} & \cdots & IDL_{ik} \\ \cdots & \cdots & \cdots & \cdots \\ IDL_{m1} & IDL_{m2} & \cdots & IDL_{mk} \end{bmatrix} \end{array} \tag{5.23}$$

具体的基于用户类标签的兴趣度计算方法如表 5.2 所示。

表5.2 基于用户类标签的兴趣度计算方法

Table. 5.2 The calculation method of interest degree based on user category label

Algorithm 5.2 基于用户类标签的兴趣度计算方法

Input：用户初始的位置坐标（X_U， Y_U），拖动后的位置坐标（$X_U^{'}$， $Y_U^{'}$），标签 L 的位置坐标为（X_L， Y_L）。

Output：用户兴趣度矩阵 $IDL_{m\times k}$

Step:

1：*Begin*

2：　*for(i in 1:m)*

3：　　*for(j in 1:k)*

4:　　$DisA \leftarrow \sqrt{(x_{u_i} - x_j)^2 + (y_{u_i} - y_j)^2}$

5:　　$DisB \leftarrow \sqrt{(x_{u_i}^{'} - x_j)^2 + (y_{u_i}^{'} - y_j)^2}$

6:　　$DisPro(l) = DisB(u_i^{'}, Lab_j) / DisA(u_i, Lab_j)$

7：　　$IDL(u_i, Lab_j) \leftarrow DisPro(l) / \max\limits_{l\in L} DisPro(l)$

8:　　进行归一户处理

9　　计算 $IDL_{i,j}$

10:　　*end　for*

11:　*end　for*

12:　*return IDL*

13:　*End*

5.3.3 用户兴趣度调整

用户的个人兴趣随着时间的变化会发生改变，这就是“兴趣漂移”现象，目前解决这一问题的主要方法有滑动时间窗口和遗忘函数法等。

本文的用户兴趣度主要是通过用户的交互行为体现的，随着时间的不断变化，用户的兴趣会出现遗忘，这和 Ebbinghaus 的原理相同，因此本文通过使用兴趣度调整函数来进行兴趣度的更新，从而通过用户交互行为更好量化用户兴趣度。在这里对以下参数进行说明。

- TimeP 表示用户的交互行为的起始时间。
- TimeO 表示用户交互行为的发生时间。
- TimeN 表示当前时间。

则兴趣度调整函数如公式 5.24 所示。

$$IAF(\Delta t_i)=\frac{(T-\Delta t_i)\cdot\xi}{T}+1,(0\le\xi\le1) \tag{5.24}$$

在公式 5.24 中：

- $\Delta t_i = TimeA - TimeP$ 表示用户交互行为发生时间距离交互行为起始的时间差
- $T = TimeN - TimeP$ 表示当前时间距离交互行为起始的时间差
- ξ 是兴趣度调整函数遗忘因子，起到调整函数的作用，其中 ξ 比较大，说明早前的用户交互行为对近期的用户兴趣度影响较大，通过设置 ξ 值的不同从而优化兴趣度调整函数。

因此，将兴趣度调整函数作为用户兴趣度的一个因子，可以调整用户交互行为随时间的推移对用户兴趣影响程度。

可以得出更新后的用户兴趣度如公式 5.25、5.26 所示。

$$IDI = IDI \bullet IAF(\Delta t_i) \tag{5.25}$$

$$IDL = IDL \bullet IAF(\Delta t_i) \tag{5.26}$$

通过公式 5.25、5.26 分别对基于用户项目的兴趣度以及基于用户类标签的兴趣度进行调整，得到更新后的用户兴趣度，从而形成用户兴趣度矩阵。

5.4 本章小结

本章主要描述基于推荐结果的用户交互行为兴趣度。首先，对用户行为进行了分析，指出目前协同过滤中存在的问题。基于此，提出基于推荐结果的用户交互行为兴趣度。

详细介绍了基于用户项目的兴趣度计算方法以及用户类标签的兴趣度计算方法，最后介绍了用户兴趣度的调整过程，从而完成整个兴趣度的计算，形成用户兴趣度矩阵。

第 6 章 实验及结果分析

6.1 实验环境

实验机器的一些配置信息如下：

（1）CPU：Intel(R)Xeon E5-2620 CPU@2.0GHz;

（2）内存：32G Bytes DDR RAM;

（3）硬盘：2T Bytes；

（4）操作系统：Windows 10；

（5）开发工具：Eclipse，R，Rstudio，Matlab；

（6）开发语言：R 语言，JAVA 语言；

在实验中，采用 R 语言对协同过滤算法进行分析，在 R 语言中有一些和协同过滤算法相关的包，如 recommenderlab，其中包括推荐算法常用的数据集 MovieLens。同时 R 语言提供了 Pearson、Spearman 等相似性计算的推荐质量指标。

6.2 实验设定

为了进行试验验证，需要采用合适的数据，本节对实验采用的 MovieLens 数据集进行了详细的说明，同时为了说明算法的有效性，本文采用了多种评价标准：MAE、准确率、召回率、F1 评分值。

6.2.1 实验数据集

本文采用协同过滤普遍使用的 MovieLens 数据集，该数据集来自于 GroupLens 研究小组提供的电影评分数据。

根据本实验的实际情况，本文选取 MovieLens 100K 数据集作为实验数据集，此数据集包含的十万条用户评分数据，其中由于数据稀疏性的问题，本文剔除评分影片数目少于 20 的用户，由于数据集包含 1682 部影片，若用户评分数目少于 20，对最终实验结果影响较大。

实验数据集包含的所有用户评分数据都在 u.DATA 文件中，具体格式为 <*UID,MID,Rating,TimeStamp*>。其中，UID 是用户的唯一标识，对应图 6.1 第一行数据；MID 是电影唯一 ID，对应图 6.1 第三行数据；Rating 是电影评分数据，范围为[1,5]，对应图 6.1 第三行数据；TimeStamp 是时间戳，对应图 6.1 第四行数据；根据实验的具体

情况，本文需要的是数据集的前三列。

```
99  4   5   886519097
178 332 3   882823437
251 100 4   886271884
81  432 2   876535131
260 322 4   890618898
25  181 5   885853415
59  196 5   888205088
72  679 2   880037164
87  384 4   879877127
290 143 5   880474293
42  423 5   881107687
292 515 4   881103977
115 20  3   881171009
20  288 1   879667584
201 219 4   884112673
13  526 3   882141053
246 919 4   884920949
```

图 6.1 u.DATA数据文件

Fig. 6.1 u.DATA data file

实验数据集包含的所有影片的详细信息在 u.ITEM 文件中，具体格式为 <*MID,Name,Time,URL,Lab*>。其中， MID 是电影唯一 ID，对应图 6.2 中的第 1 部分；Name 是影片名称，对应图 6.2 中的第 2 部分；Time 是影片上映时间，对应图 6.2 中的第 3 部分；URL 是影片网址，对应图 6.2 中的第 4 部分；Lab 是影片类别标签，对应图 6.2 中的第 5 部分；

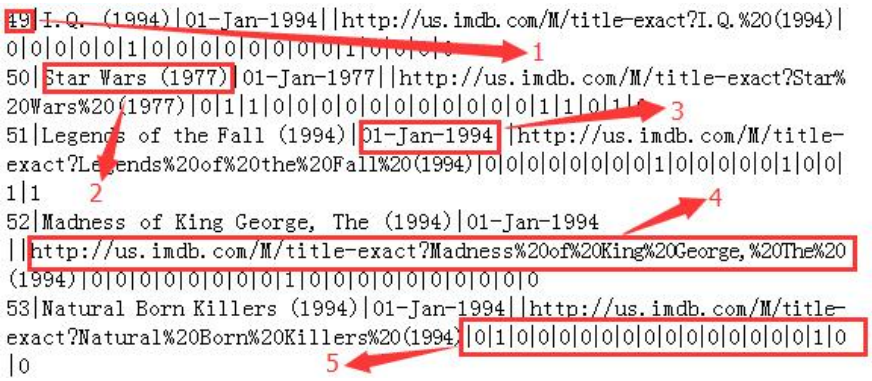

图 6.2 u.Item影片信息文件

Fig. 6.2 u.Item movie information file

“稀疏等级”[40-42]是衡量数据系数程度的有效指标，对于用户评分矩阵 $R_{m\times n}$ 的稀疏等级可用公式6.1进行描述。

$$s=1-\frac{number\ of\ R_{i,j}\neq\varnothing}{number\ of\ R_{i,j}},1\le i\le m,1\le j\le n \tag{6.1}$$

根据稀疏等级公式： $s=1-100000/(943*1682)=0.9370$ ，因此，本实验数据集的稀疏性很高。在本实验中将测试集与训练集按照20%和80%的比例进行分配，形成u.test和u.base。

6.2.2 实验评价方法推荐

推荐性能的评价常用以下几个指标衡量[43]：

（1）平均绝对误差（MAE）

MAE是最常用的评价标准之一，它主要把用户的预测评分与实际评分的差异作为算法质量好坏的一个标准。其公式如6.2所示:

$$MAE = \frac{1}{I_T}\sum_{(u,i)I_r} | p_{ui} - r_{ui} | \tag{6.2}$$

其中， p_{ui} 是预测评分， r_{ui} 是实际评分。

（2）准确率（Precision）

在推荐系统中，准确率是指用户的推荐结果中用户真正需要的结果在整个推荐结果中的比例。其公式如6.2所示。

$$P = \frac{N_{rs}}{N_s} = \frac{N_{rs}}{N_{rs} + N_{is}} \tag{6.3}$$

其中， N_{rs} 是用户真正需要的结果数目， N_{is} 是对用户没有贡献推荐结果数目。

（3）召回率（Recall）

召回率是指是推荐算法提供给用户的相关推荐项目与所有可能相关的项目之间的比值，其公式如6.3所示。

$$R = \frac{N_{rs}}{N_r} = \frac{N_{rs}}{N_{rs} + N_{rn}} \tag{6.4}$$

其中， N_{rs} 是用户真正需要的结果数目， N_r 是可能相关的项目数。

（3）F1评分

F值是对Precious以及Recall的综合度量，如公式6.4所示。

$$F_1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times N_{rs}}{N_s + N_r} \tag{6.5}$$

6.3 实验分析

本节通过设计不同的实验方案，验证本文提出的交互式协同过滤推荐方法中URDE、UIBS方法的可行性与有效性。首先设计实验的过程，然后对实验对比结果进行分析总结。

6.3.1 URDE 方法对协同过滤算法性能的影响

（1）实验过程

本实验的目的是考察本文提出的基于用户评分差异度信息熵的相似性计算（URDE）方法在用户评分数据稀疏的情况下相似性计算的精度，进而表现为推荐结果的准确性。通过6.2.1的分析看出实验使用的MovieLens数据集的稀疏等级为0.9370，属于稀疏性比较严重的数据，符合实验的要求。

本实验首先需要对实验数据集进行处理，通过创建数据模型DataModel，对数据集的UID、MID、Rating进行提取，形成用户评分矩阵 $R_{m\times n}$ 。考虑用户自身评分的偏好不同，针对用户评分矩阵中用户参与的评分项目，这里通过用户评分与用户整体评分的均值相减取整，处理后的数据加上调节参数 δ 使之评分值为正数，由于数据集的评分采用五星级标准，因此取 δ 为4，之后考虑将评分减去评分最小值使用户评分矩阵的结果值在[1,5]之间。

接下来计算用户间相似性，这里采用的方法是本文提出的URDE方法。首先得出用户间的共同评分项目L，在此基础上计算共同评分项目的差异度，存入集合*diffScore*中，针对本实验数据，差异度分为九个等级，分别是{-4,-3,-2,-1,0,1,2,3,4}分别计算其在集合*diffScore*出现的概率，之后通过加权差异度信息熵计算公式计算用户间相似性，考虑用户相似性的特点，需要对其进行归一化，使其结果在[0,1]中。

接下来定义通过用户近邻算法，分别设置 $K=\{10,15,20,25,30,35,40,45,50,55,60\}$，计算用户近邻，最后调用推荐算法预测用户评分，6.2.1节中将用户训练集与测试按照20%:80%比例进行分配，这里通过计算用户预测评分与测试集中的用户推荐项目实际评分的差来计算此算法的MAE值，作为衡量算法精度的一个标准。之后给出对推荐结果进行TOP-N推荐，这里取N=30，根据测试集与实际TOP-N推荐结果计算此算法的准确率、召回率以及F1评分。

（2）对比实验结果分析

为了验证本文提出的URDE方法的有效性，本节使用Pearson相似性计算方法、Spearman相关计算方法、余弦相似性计算方法和进行对比，通过分别设置用户近邻的数目为 $K=\{10,15,20,25,30,35,40,45,50,55,60\}$，对Pearson相似性计算、Spearman相关、Cosine相似性计算以及本文的URDE方法，通过u.base得到的预测评分与u.test中的实际评分计算的预测评分，计算其MAE值，通过对比，其结果如图6.3所示。

由图6.3可知，在用户近邻数比较小的情况下本文提出的URDE算法的MAE值并不是特别理想，几种相似性方法的MAE值差别不大。随着近邻数目的增多，本文提出的URDE算法效果明显低于其它三种相似性计算方法。

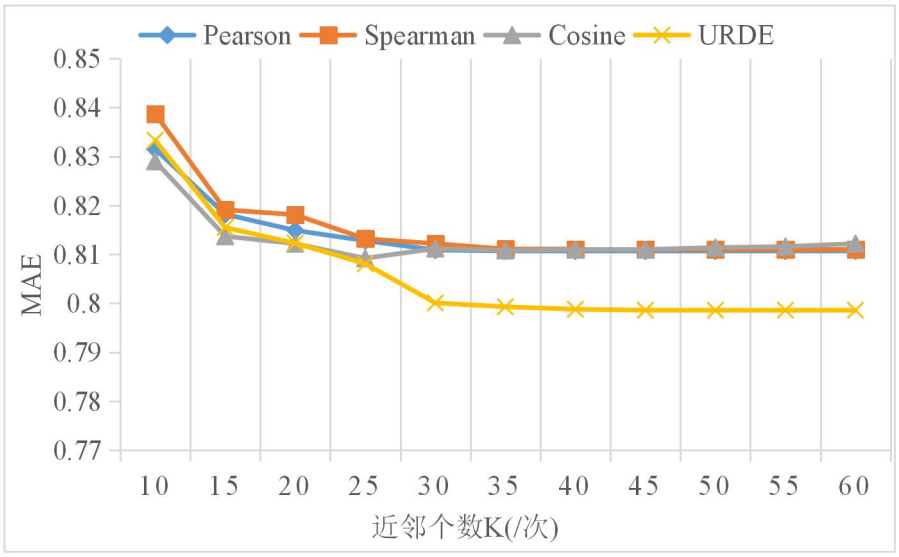

图 6.3 四种相似性计算方法在不同K值情况下的推荐结果MAE值对比图

Fig 6.3 Comparison of recommended results MAE values for four similarity calculation methods at different K values

同时本实验对比了本文提出的URDE算法与Pearson、Spearman、Cosine相似性计算方法的准确率、召回率以及F_1评分，对比结果如图6.4所示。

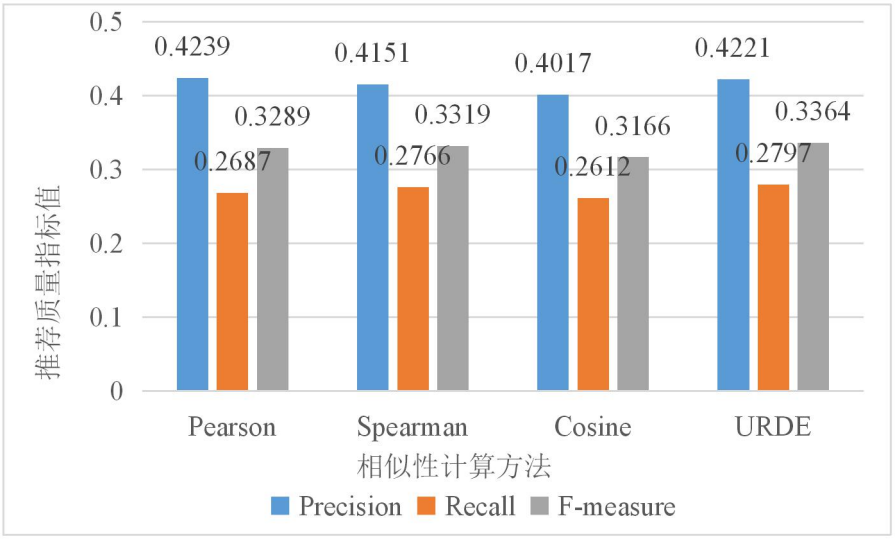

图 6.4 四种相似性计算方法推荐质量对比图

Fig. 6.4 Comparison of four similarity calculation methods in the recommended quality

由图6.4可知，本文提出的URDE算法准确率稍低于Pearson相似性计算，召回率以及F1测试值均比较好。由于图中四种方法的Recall相对比较低，可能是由于数据集中用户数是943，而本实验最大的近邻数是60造成的，同时实验数据集的大小也对其造成一定影响。

6.3.2 UIBD 方法对协同过滤算法性能的影响

（1）实验过程

本实验的目的是考察本文提出的基于用户交互行为兴趣度的相似性计算（UIBD）方法对最终推荐结果性能的影响。本实验室是在6.3.1的基础上进行的，通过基于用户评分差异度信息熵的相似性计算方法得出的推荐结果，在其基础上用户对其交互，用户的交互行为代表用户的兴趣信息。由于数据规模的问题，这里抽样200个用户进行试验，同样测试集采用MovieLens 20%的数据。

本实验首先对用户交互行为的处理，根据用户对象交互对象的不同，这里将用户交互行为分为基于用户项目的交互行为，以及基于用户类标签的交互行为。

其中，基于用户项目的交互行为分为用户访问推荐列表条目行为（UAIB）、用户观看影片行为（UWMB）、用户操作鼠标行为（UOMB）。对于 UAIB，通过用户访问某一推荐列表条目的次数表达其兴趣，记录在 AFre 矩阵中。对于 UWMB，在 MovieLens 数据集的 u.Item 中保存影片对应的 URL 信息，用户对其感兴趣的影片会进行观看，这里通过记录用户影片观看时常占比来计算其对应兴趣信息，存储在 WPro 中，对于 UOMB，通过 ML 记录鼠标左键的点击次数，通过 MR 记录鼠标右键的操作次数，通过 PD 记录按下 PageDown 的次数，通过 PU 记录按下 PageUp 的次数，通过 CC 记录光标键在产生的滚动次数，通过 MC 记录鼠标滚轮产生的滚动次数，最后结果存在 MFre 矩阵中。之后通过基于用户项目的兴趣度计算方法计算其各自对应的兴趣度，形成兴趣度矩阵。兴趣度矩阵是用户对项目的兴趣度值。在用户最小交互行为组合兴趣度中，根据经验值，这里分别对 UAIB、UWAB 以及 UOMB 三种交互行为兴趣度的权重 λ、η、β 设为 0.25,0.55,0.2，计算得出基于用户项目的兴趣度矩阵 IDI。接下来使用基于用户项目兴趣度的相似性计算方法计算用户间相似性矩阵 ItSim。

基于用户类标签的交互行为是指用户通过调整与不同类标签的距离来表达对其感兴趣程度 。这里根据 MovieLens 数据集的 u.Item 将影片类标签分为 18 种。

用户对应一个初始位置，记录其坐标值，同时每个类别标签对应一个坐标值。用户通过鼠标拖动调整其与不同类标签的远近，记录其位置，这里通过用户距离占比来表示用户的兴趣度，通过基于用户类标签的兴趣度计算方法分别计算用户对各个类标签的兴趣度，形成用户兴趣度矩阵 IDL，接下来使用基于用户类标签兴趣度的相似性计算方法计算用户间相似性，得出用户相似性矩阵 *LabSim*。通过使用 UIBD 方法计算用户间相似性矩阵 *UIBDSim*。

分别设置近邻数目 $K=\{10,15,20,25,30,35,40,45,50,55,60\}$ 计算用户近邻，最后预测用户评分，这里通过计算用户预测评分与测试集中的用户推荐项目实际评分的差来计算此算法的MAE值，作为衡量算法精度的一个标准。之后给出对推荐结果进行TOP-N推荐，

这里取N=10，根据测试集与实际TOP-N推荐结果计算此算法的准确率、召回率以及F1评分。

（2）实验结果分析

通过分别设置用户近邻的数目为$K=\{10,15,20,25,30,35,40,45,50,55,60\}$，对Pearson相似性计算、Spearman相关、Cosine相似性计算以及本文的UIBD方法，通过u.base得到的预测评分与u.test中的实际评分计算的预测评分，计算其MAE值，通过对比，其结果如图6.5所示。

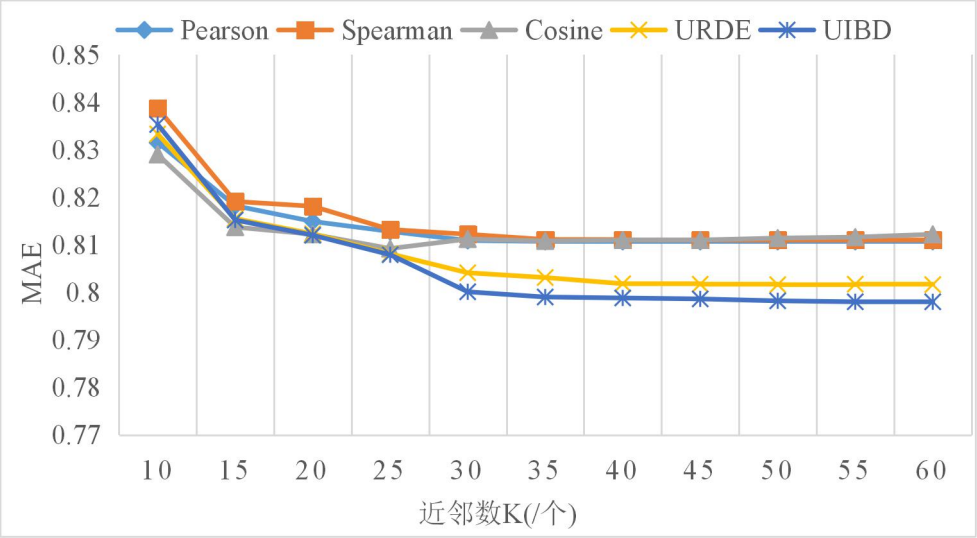

图 6.5 四种相似性计算方法在不同K值情况下的MAE值对比图

Fig 6.5 Comparison of MAE values for four similarity calculation methods at different K values

由图6.5可知，本文提出的基于用户交互行为兴趣度的相似性计算方法在近邻数较大时，MAE值相对于其它三种相似性计算方法效果比较明显，在近邻数比较小时，几种算法MAE差别不大。

同时本实验对比了本文提出的基于用户交互行为兴趣度的相似性计算方法与Pearson、Spearman、Cosine相似性计算方法的准确率、召回率以及F1测试值，对比结果如图6.6所示。

由图6.6可以看出，本文提出的基于用户交互行为兴趣度的相似性计算方法，在推荐结果的准确率召回率以及F1值相对于其它三种相似性计算方法效果均比较好。可见用户的参与对最终的结果有积极地影响。

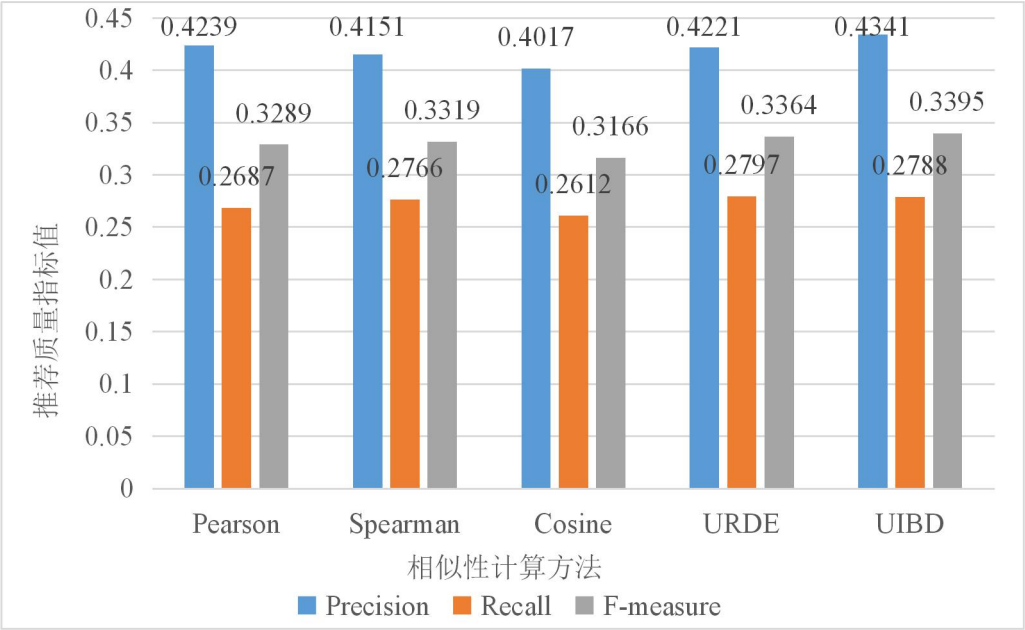

图 6.6 四种相似性计算方法在推荐质量对比图

Fig. 6.6 Comparison of four similarity calculation methods in the recommended quality

6.4 本章小结

本章针对基于用户评分差异度信息熵的相似性计算方法以及基于用户兴趣度的相似计算方法进行实验验证。首先进行了实验数据集的介绍，给出实验结果的评价标准。设计实验分别对两个方法的MAE、准确率、召回率以及F1值进行计算并与其它相似性计算方法进行对比，结果表明本文提出的算法能够提高推荐精度。

第 7 章 总结与展望

7.1 论文工作总结

本文旨在解决协同过滤的数据稀疏性问题、用户推荐反馈问题以及基于用户反馈的推荐优化问题，基于此，提出一种交互式协同过滤推荐方法，并构建了交互式协同过滤推荐框架以及交互式协同过滤推荐实现过程。在此基础上，对其中的相似性计算问题、兴趣度计算问题进行了深入研究，本文的主要工作包括：

（1）构建交互式协同过滤的框架

本文在传统协同过滤的基础上提出了交互式协同过滤推荐方法，构建了交互式协同过滤推荐框架，并给出具体的实现过程。该方法主要包括三个主要部分：数据预处理、用户相似性的计算、用户兴趣度的计算。数据预处理对原始的用户评分数据进行提取形成用户-项目评分矩阵；用户相似性计算主要缓解传统协同过滤算法中的数据稀疏性问题，以及给予反馈的推荐优化问题；兴趣度的计算主要是解决用户推荐反馈问题。

（2）提出基于用户评分差异度信息熵的相似性计算方法

相似性计算是整个协同过滤推荐算法的核心，本文提出一种基于用户评分差异度信息熵的相似性计算方法。该方法对由于用户自身评价偏好、用户间评分差异、评分数据的稀疏性引起的推荐精度不高问题有很好的效果。

（3）提出基于推荐结果的用户交互行为兴趣度计算方法

针对用户的不同交互行为本文提出基于推荐结果的用户交互行为兴趣度计算方法。该方法将用户的行为分为基于项目的行为以及基于类标签的行为：基于项目的行为主要是指用户与推荐列表之间的交互行为，通过对用户交互行为兴趣度的计算形成用户兴趣度矩阵，通过用户最小交互行为组合兴趣度计算方法计算用户项目兴趣度矩阵用户项目相似性的计算；基于类标签的行为主要是指用户与分类标签之间的交互行为，通过欧氏距离计算得出用户类标签兴趣度矩阵，用于用户类标签相似性的计算。该方法将用户的交互行为转化为用户的兴趣度矩阵，并应用到协同过滤的推荐过程中，解决了用户推荐反馈问题。

（4）提出基于用户交互行为兴趣度的相似性计算方法

针对用户项目相似性与用户类标签相似性本文提出基于用户项目兴趣度的相似性计算方法以及基于用户类标签兴趣度的相似性计算方法，分别针对用户项目兴趣度矩阵和用户类标签兴趣度矩阵进行计算得到对应的用户相似性，该方法分别针对用户项目兴

趣度矩阵和用户类标签兴趣度矩阵进行相似性的计算，从而优化相似性计算的方法，解决基于用户反馈的推荐优化问题。

7.2 下一步研究工作展望

虽然本文提出的算法对于解决数据稀疏性关键问题的效果进行了改进，取得了一定效果，但是受制于实验环境、实验手段、实验数据等因素，仍然有许多地方可以进行改进主要有以下方面：

（1）构建更好的兴趣度计算方法。

本文的交互行为主要分为基于项目的交互行为以及基于类标签的交互行为，只选取了其中比较重要的交互行为进行了考虑，在未来的工作中，可以考虑对更多的用户交互行为，从而提供更好的交互式推荐。同时在获取交互行为的基础上兴趣度的计算也是一个关键，可以在未来的工作中对此进行进一步研究。

（2）考虑用户兴趣的更新，建立相关的兴趣度更新模型

用户兴趣会随着时间的推移出现一定程度的变化。因此，一个好的用户兴趣度更新模型可以在一定程度上提高最终推荐的精度。本文主要考虑通过滑动时间窗口的方法进行兴趣度的更新，主要考虑了时间，未来的工作可以把更多的因素纳入考虑范围之内。

协同过滤推荐技术的研究早已是一个热点，随着研究的不断深入，上述问题会得到更好的解决。

参考文献

[1] Sobotta N. A Systematic Literature review on the relation of information technology and information overload[J]. 2016:858-867.

[2] Campos R, Dias G, Jorge A M, et al. Survey of Temporal Information Retrieval and Related Applications[J]. Acm Computing Surveys, 2015, 47(2):1-41.

[3] Kobayashi I, Saito M. A study on an information recommendation system that provides topical information related to user's inquiry for information retrieval[J]. New Generation Computing, 2007, 26(1):39-48.

[4] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence, 2009, 2009(12).

[5] Feng Z J, Xian T, Feng G J. An Optimized Collaborative Filtering Recommendation Algorithm[J]. Journal of Computer Research & Development, 2004, 41(10):1842-1847.

[6] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12):61-70.

[7] Su J H, Yeh H H, Yu P S, et al. Music Recommendation Using Content and Context Information Mining[J]. Intelligent Systems IEEE, 2010, 25(1):16-26.

[8] Yin C, Zhang H, Xiang J, et al. A New Mobile Recommendation Algorithm Based on Statistical Theory[C]. In: International Conference on Advanced Information Technology and Sensor Application. 2015:96-99.

[9] Qi H, Ming C, Xiao M. A Personalized Resource Recommendation System Using Data Mining[C].In: International Conference on E-Business and E-Government. IEEE Computer Society, 2010:5365-5368.

[10] Krzywicki A, Wobcke W, Kim Y S, et al. Collaborative Filtering for people-to-people recommendation in online dating: Data analysis and user trial[J]. International Journal of Human-Computer Studies, 2015, 76(C):50-66.

[11] Jing H, Liang A C, Lin S D, et al. A Transfer Probabilistic Collective Factorization Model to Handle Sparse Data in Collaborative Filtering[C].In: IEEE International Conference on Data Mining. IEEE, 2015:250-259.

[12] Bobadilla, Jes&#, Ortega F, Hernando A, et al. A collaborative filtering approach to

mitigate the new user cold start problem[J]. Knowledge-Based Systems, 2011, 26:225-238.

[13] Wu Y H, Tan X Q. A real-time recommender system based on hybrid collaborative filtering[C]. In: International Conference on Computer Science and Education. IEEE, 2010:1909-1912.

[14] Jing H, Liang A C, Lin S D, et al. A Transfer Probabilistic Collective Factorization Model to Handle Sparse Data in Collaborative Filtering[C].In: IEEE International Conference on Data Mining. IEEE, 2015:250-259.

[15] Cui H, Zhu M. Collaboration filtering recommendation optimization with user implicit feedback[J]. Journal of Computational Information Systems, 2014, 10(14):5855-5862.

[16] Zhong Z, Yong S, Yue W, et al. An improved collaborative filtering recommendation algorithm not based on item rating[C].In: IEEE, International Conference on Cognitive Informatics & Cognitive Computing. IEEE, 2015:230-233.

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

[18] Basu C, Hirsh H, Cohen W. Recommendation as classification: using social and content-based information in recommendation[C].In: Fifteenth National/tenth Conference on Artificial Intelligence/innovative Applications of Artificial Intelligence. American Association for Artificial Intelligence, 1998:714-720.

[19] Sandvig J J, Mobasher B, Burke R. Robustness of collaborative recommendation based on association rule mining[C] .In: ACM Conference on Recommender Systems, Recsys 2007, Minneapolis, Mn, Usa, October. 2007:105-112.

[20] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence, 2009, 2009(12).

[21] Schafer J B, Dan F, Herlocker J, et al. Collaborative Filtering Recommender Systems[J]. Acm Transactions on Information Systems, 2004, 22(1):5-53.

[22] Goldberg D, Nichols D, Oki BM, et al. Using collaborative filtering to weave an information tapestry. Communications of the ACM[J]. December, 1992, 35(12): 61-70.

[23] www.grouplens.org.

[24] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C].In: Proceedings of the 14th Conference on Uncertainty in

Artificial Intelligence (UIA'98), 1998, 43-52.

[25] Y Chuan, X Jie-ping. Recommendation algorithm combining the user-based classified regression and the item-based filtering[C] In: Processing of the International Conference on Electronic Commerce, Proceedings-the new E-commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet,2006,574-578.

[26] Arwar B, Karypls G, Konstan J, etal. Item-based collaborative filtering recommendation algorithms [C]. In: Proceedings of the 10th International World Wide Web Conference, 2001.

[27] Youngki Park,Sungchan Park,Woosung Jung,Sang-goo Lee. Reversed CF: A fast collaborative filtering algorithm using a k -nearest neighbor graph[J]. Expert Systems With Applications . 2015 (8).

[28] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C].In: International Conference on World Wide Web. ACM, 2001:285-295.

[29] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UIA'98), 1998, 43-52.

[30] Chen Y H,George E I.A Bayesian Model for Collaborative Filtering[C].In: Process of the 7th International Workshop on Artificial Intelligence and Statistics,1999:56-60.

[31] Li B, Yang Q, Xue X. Transfer learning for collaborative filtering via a rating-matrix generative model[C].In:International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June. 2009:617-624.

[32] Chan N N, Gaaloul W, Tata S. A Web Service Recommender System Using Vector Space Model and Latent Semantic Indexing[C]. In: IEEE International Conference on Advanced Information NETWORKING and Applications. IEEE Computer Society, 2011:602-609.

[33] 赵亮,胡乃静,个性化推荐算法设计川.计算机研究与发展,2002.39(8):P986-991.

[34] Insuwan W, Suksawatchon U, Suksawatchon J. Improving missing values imputation in collaborative filtering with user-preference genre and singular value decomposition[C]. In: International Conference on Knowledge and Smart Technology. 2014:87-92.

[35] S.J.D. Phoenix. Elements of Information Theory[J]. 1992, 39(7):1600-1601.

[36] 曾春,邢春晓,周立柱.个性化服务技术综述[J].软件学报,2002,13(10):1952-1961.

[37] M Claypool,P Le,M Waseda,et al. Implicit Interest Indicators[A]. Proc of the ACM Intelligent User Interfaces Conf(IUI)[C]. 2001. 33-40.

[38] T P Liang,H J Lai.Discovering User Interests from Web Browsing Behavior:An Application to Internet News Services[A]. Proc of the 35th Hawaii Int' l Conf on System Sci-ences[C]. 2002.

[39] B M Sarwar.Sparsity, scalability, and distribution in recommender systems[D]. Minneapolis, MN: University of Minnesota, 2001.

[40] B Sarwar,G Karypis,J Konstan, et al. Item-based collaborative filtering recommendation algorithms[C]. In: Proceediings of the 10th International Conference on World Wide Web. New York: ACM Press, 2001. 285-295.

[41] B Sarwar,G Karypis,J Konstan, et al. Analysis of recommendation algorithms for E-commerce[C]. In: Proceedings of the 2nd ACM Conference on Electronic Commerce. New York: ACM Press, 2000. 158-167.

[42] Herlocker J L,Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. Acm Transactions on Information Systems, 2004, 22(1):5-53.

致 谢

时间总在不经意间偷偷溜走，一转眼，两年半的研究生阶段马上结束，我的人生也将踏上新的征程，去迎接新的机遇与挑战。回想两年半的时间，心里满满的是不舍与感激，不舍朝夕相处的老师同学，不舍两年半的校园生活。在此我向所有关心帮助过我的老师同学表示衷心的感谢。

感谢我的导师张斌教授。在研究生期间，张老师对我悉心指导，他对待学生认真负责的态度，对待学术严谨的精神，深深影响着我。两年半以来，每次与张老师进行学术交流都使我受益匪浅，使我成为一个具有严谨的逻辑思维，能够独立解决问题的研究生。我会谨记您的教诲，不断激励自己。

感谢杨雷老师在我研究生期间对我的指导，杨老师对待项目钻研的精神深深影响着我。与杨老师针对项目问题一次次地讨论使我的技术能力得到很大提升，能够独立完成部分项目模块的开发设计工作。杨老师对待工作的热情也深深地感染我。

感谢代钰老师。代老师在我的论文撰写期间对我提供很大的帮助，代老师对问题深入的理解，清晰的思路，使我豁然开朗，并鼓励我主动去思考与创新，使我成为一名更合格的硕士研究生。

感谢实验室的赵世成、苏道磊、鲁亚楠、任凯丽、吴乌日古木拉、张翔同学，在与他们共同学习生活过程中使我收获很多，感谢实验室的孟政宇、张俊英、张通等师弟师妹对我论文的检查，感谢吴嘉轩、王鹏师兄。感谢我的室友朱坤鹏、雷新会、关博馨、邓鸿兵。

最后感谢评阅我论文的专家、学者和老师，感谢您在百忙之中对我论文的评阅。