分类号_____ 密 级_____

UDC _____

学 位 论 文

协同过滤优化算法的研究与实现

作 者 姓 名： 陈玲玲

指 导 教 师： 王大玲 教授

东北大学信息科学与工程学院

申请学位级别： 硕士 学 科 类 别 ： 工学

学科专业名称： 计算机软件与理论

论文提交日期： 2008 年 6 月 15 日 论文答辩日期： 2008 年 7 月 1 日

学位授予日期： 2008.7 答辩委员会主席： 申德荣

评 阅 人 ： 杨晓春 . 李晓光

东 北 大 学

2008 年 6 月

A Thesis in Computer Software and Theory

Research and Implementation for Collaborative Filtering Optimization Algorithm

by Chen Lingling

Supervisor :Professor Wang Daling

Northeastern University

June 2008

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，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他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：陈玲玲

日　　期：2008.6.20

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：陈玲玲　　导师签名：

签字日期：2008.6.20　　签字日期：2008.6.20

协同过滤优化算法的研究与实现

摘 要

随着互联网和电子商务的发展，电子商务推荐系统逐渐成为一个重要研究内容，得到了研究者越来越多的关注。其中，协同过滤推荐技术是目前推荐系统中应用最早和最为成功的技术之一，是个性化推荐领域重点研究的课题。

本文通过分析了协同过滤推荐技术目前存在的问题，指出随着电子商务系统用户数目和商品数目的日益增加，整个项目空间上用户评分数据极端稀疏，传统的相似性度量方法没有强调项目所属类别对相似性计算的影响，因而计算结果不够准确。针对该问题，提出了项目类型信息参与相似性计算的思想。将此思想分别应用于基于项目协同过滤算法和基于用户协同过滤算法中，前者使用项目类型矩阵计算类型部分，后者使用由项目——类型矩阵与用户评分矩阵得到的用户——项目类型矩阵进行计算类型部分，并将其与各自相应的传统相似性计算结果线性结合一并作为项目间和用户间的相似性。实验结果表明，在基于项目和基于用户协同过滤算法中，该方法不同程度地提高了预测的精确度。

本文还就传统协同过滤算法无法反映用户对不同类项目的关注度的不同问题，提出一种改进的基于用户的协同过滤算法。该算法利用组合推荐方法思想，结合了基于项目和基于用户协同过滤算法。该算法以基于用户协同过滤算法为主体，使用基于项目协同过滤算法得出待预测项目的邻居项目，对基于项目协同过滤算法产生的目标用户的邻居集合进行再次选择，它能考虑到用户在不同类项目的兴趣差异，找到针对每个类项目与用户“真正”的邻居用户。实验结果表明，算法能有效避免传统方法的弊端，提高预测精度，从而提高了协同过滤系统的推荐质量。

关键词：推荐系统；协同过滤；项目；用户；优化；组合推荐

Research and Implementation for Collaborative Filtering Optimization Algorithm

Abstract

With the development of Internet and E-commerce, the recommender system has gradually become an important research field of E-commerce technology, and attracts many researchers' attention. Collaborative filtering technology is one of earliest applied and most successful technologies in recommender system. And it is the main research issues in the field of personalized recommendation, and in this paper we focus our research on it.

In this thesis, it is pointed out that with the development of E-commerce, the magnitudes of users and commodities grow rapidly, which resulted in the extreme sparsity of user rating data by analyzing the problems that exist in today's collaborative filtering technology. For emphasizing the impact of item's genres on similarity computing, the traditional similarity measure methods work inaccurately in this situation. For this problem, an idea of making item's genre information take part in similarity computing is proposed. It is applied in item-based and user-based collaborative filtering algorithms respectively. The former uses the matrix of item and item's genre for genre part, the latter uses the matrix of user and item's genre gained from the matrix of user rating and the matrix of item and item' genre for that part, then it is combined with each corresponding original similarity via linear approach to become similarity between items and between users. Experiments show it increases accuracy of prediction at different levels both in item-based and user-based collaborative filtering.

And then a modified user-based collaborative filtering algorithm is proposed for solving the problem that the traditional collaborative filtering can't reflect the difference of user' attention to different kinds of items. The algorithm combines item-based and user-based collaborative filtering according to combined recommendation. This algorithm regards user-based collaborative filtering as the main body, using neighbors of item to be predicted produced from item-based collaborative filtering to select nearest neighbors of active user got from user-based collaborative filtering again. It is able to take the difference of user's interest in different kinds of items and find 'real' neighbors of active users regarding every kind of items. Experiments show it can effectively avoid the shortcomings of traditional methods and improve the accuracy of prediction, thereby enhancing the collaborative filtering system on the recommendation quality.

Keywords: recommender system; collaborative filtering; item; user; optimization; hybrid recommendation

目 录

第一章 绪 论

1.1 研究背景

近年来，随着互联网的普及和 web 技术的日新月异促进了电子商务的快速发展，电子商务的日益繁荣改变了传统的贸易行为，它的逐步建立和完善使传统的商务运作摆脱了已有规则的束缚，对相关的商业形态、交易形式、流通方式以及营销方式等都产生的巨大的影响。

在这一潮流趋势下，企业和用户都面临了新的形势。对企业而言，电子商务为企业发展提供了新的商业入口，同时也提供了大量的产品信息，创造了更多的商业机会。全球信息化使企业原本拥有的传统的地域性优势和由于信息不对称带来的信息优势在很大程度上被削弱，同时又使企业能够获得更广阔的原料来源和市场选择。网络也为企业提供了廉价、方便快捷、手段多样的市场调研环境和营销手段，可以更有效地与用户接触交流。对用户而言，电子商务为他们提供了前所未有的产品选择空间和购物便利。在这个能轻松得到各种资讯的信息海洋里，用户却从一开始的惊喜，变得有些无所适从。如何在茫茫的信息海洋中找到自己要的东西，已经成为令企业和个人用户头疼的问题。虽然搜索引擎、信息检索等能为用户提供一定的帮助，但是它为每个用户提供的服务都是一样的，而且反馈的信息量也比较大[1]。不能从根本上帮助用户解决这个问题。产品种类的极大丰富使得用户的购买目的从单纯的满足对物质的需要更多地转变为体现个性特征和满足个性化需求。

因此，为满足用户和企业共同的迫切需要，重视用户的个体需求，致力于满足不同用户的不同偏好的电子商务个性化推荐系统(Personalized Recommendation System for E-Commerce)应运而生。目前，几乎所有大型的电子商务系统。如 Amazon、eBay、阿里巴巴等，都不同程度的使用了各种形式的推荐系统。

为了保证推荐系统实时性条件下产生相对精确的推荐，研究者提出了多种不同的推荐算法，如协同过滤技术、聚类技术、关联规则技术、Horting 图技术等[2]，其中协同过滤技术，构成了现有电子商务个性化推荐系统的基础。协同过滤的出发点是：兴趣相近的用户可能会对同样的东西感兴趣。所以只要维护关于用户喜好的数据，从中分析得出具有相似口味的用户，然后就可以根据相似客户的意见来向其进行推荐。另一种可能的出发点是：用户可能较偏爱与其已购买的东西相类似的商品，可以根据用户对各种东西的评价来判断商品之间的相似程度,然后推荐与用户兴趣最接近的那些商品。前一种思路以客户与客户之间的关系为中心,而后一种思路则以项目与项目之间的关系为着眼点。

协同过滤在国内外尤其是国外各个大型电子商务网站的推荐系统中都得到了不同

程度的应用，像 GroupLens：过滤网上新闻的系统；Ringo：推荐音乐的系统；Video Recommender 和 MovieLens：推荐电影的系统，Jeste：推荐笑话的系统等。其后，该技术的商业应用也不断的扩大，如 Amazon.com、CDNow.com 等，都使用了协同过滤技术向顾客推荐产品。因而协同过滤推荐技术是个性化推荐领域重点研究的课题。

1.2 本文主要研究内容

本文通过对电子商务推荐系统中的协同过滤推荐技术以及其存在的问题等的研究，分析电子商务推荐中的实际问题与需求，针对协同过滤推荐算法的一些缺陷，并结合当前个性化服务推荐技术的前沿，主要进行以下工作：

在传统协同过滤基于用户评分矩阵的计算的基础上，考虑到用户评分数据稀疏性和项目分类信息对项目以及对用户相似性的影响，引入项目类型信息参与传统协同过滤中相似度的计算。本文采用线性结合方式将通过用户评分矩阵计算的传统相似度和通过项目类型信息计算的相似度结合一并作为事物的相似度。本文分别就此想法在基于项目协同过滤和基于用户协同过滤中的具体的实现进行了说明，并进行了相应的实验。通过实验验证采用本文提出的方法计算相似性，它在基于项目和基于用户的协同过滤算法中的应用对推荐的质量都有不同程度的改善和提高。

本文还就传统的协同过滤推荐算法无法反映用户对不同类的项目的偏好差异，使得推荐时缺少个性，推荐质量不高等问题，利用组合推荐思想，结合基于项目和基于用户二种协同过滤推荐算法而提出了改进的基于用户协同过滤算法。该算法利用基于项目协同过滤产生的待预测项目的最近邻居集合对基于用户协同过滤产生的目标用户的最近邻居用户进行再选择，淘汰没有对待预测项目的最近邻居集合中任何项目评分的用户，找到在每一类项目上真正与目标用户相似的用户，以期提高预测评分精度和改善推荐质量。通过进行一系列实验验证算法能否达到预期效果，进一步提高推荐系统的推荐质量和体现推荐的个性化。

1.3 本文的组织结构

本文各章的结构安排如下：

第一章为绪论，主要介绍本课题的研究背景、本文研究内容。

第二章为相关技术理论与技术，首先简单介绍了电子商务推荐系统概念构成及其作用，并进一步介绍了目前电子商务推荐系统中的主要的协同过滤技术。接着着重介绍协同过滤的基本思想、实现过程、现有的协同过滤推荐系统和分类，认识和了解协同过滤技术，接着重点介绍基于协同过滤的推荐系统面临的挑战，研究和分析协同过滤算法存在的问题及目前比较典型的解决问题的方法，并对它们优缺点进行了分析。

第三章为基于项目的协同过滤的类型优化算法，这一章通过分析项目类型信息对项

目相似性度量的重要影响，提出在传统基于项目协同过滤推荐算法只利用用户评分数据的基础上引入项目类型信息，利用该信息参与计算项目间的相似度的想法。并通过实验测试其是否能改善推荐质量。

第四章为基于用户协同过滤的类型优化算法，通过分析传统基于用户协同过滤推荐算法中的不足，说明项目类型对探究用户深层兴趣以及用户相似性度量中的作用。提出在基于用户协同过滤推荐算法计算用户相似步骤中性引入项目类型信息，并通过实验验证算法的有效性。

第五章为改进的基于用户的协同过滤算法。该章首先介绍组合推荐思想和相应研究成果，接着分析了传统协同过滤算法存在的不足以及相关工作，提出改进的基于用户协同过滤算法。该算法通过结合基于项目协同过滤算法，对传统的基于用户协同过滤算法产生用户的最近邻居进行再选择，以期提高预测水平，并通过实验验证，它能提高推荐质量，使推荐更具个性化。

最后是结论与展望，总结全文内容并提出进一步研究的方向。

第二章 相关理论与技术

2.1 电子商务推荐系统简介

随着互联网的普及和电子商务的高速发展，互联网已成为一个分布广泛的全球性信息服务中心和人们获取信息的一个重要途径。然而随着网络信息的快速增长，人们却常常面临“信息爆炸”的尴尬处境，不得不花费大量的时间去搜索、浏览自己需要的信息。在电子商务的虚拟环境下，商家所提供的商品种类和数量非常多，用户不可能通过一个小小的计算机屏幕一眼就知道所有的商品，用户也不可能象在物理环境下那样检查挑选商品。因此，需要商家提供一些智能化的选购指导，根据用户的兴趣爱好推荐用户可能感兴趣或是满意的商品，使用户能够很方便地得到自己所需要得到的商品。而且，从现实经验来看，用户的需求经常是不明确的、模糊的，可能会对某类商品有着潜在的需求，但并不清楚什么商品能满足自己的模糊需求。这时，如果商家能够把满足用户模糊需求的商品推荐给用户，就可以把用户的潜在需求转化为现实的需求，从而提高产品的销售量。在这种背人们对信息个性化的要求越来越高。为了满足用户个性化的需求，电子商务推荐系统应运而生。

电子商务推荐系统一正式的定义是 Resnick & Varian 在 1997 年给出的：“它在电子商务系统中向客户提供商品信息和建议，帮助客户决定购买何种商品，模拟销售人员向客户推荐商品完成购买的过程”，现在这个定义已被广泛引用。推荐系统推荐何种商品是在电子商务网站整体商品的购买情况、客户的人数统计或者对客户购买的历史记录上进行分析产生的。广义上讲，这些因素的考虑使电子商务具有了个性化的色彩，而且对于不同的客户，具有推荐系统的电子商务网站表现出了一定的自适应性。

2.1.1 电子商务推荐系统的构成

一般，电子商务推荐系统都可分为三个模块输入模块、推荐算法模块、输出模块。

(1) 输入模块 主要负责推荐系统数据源的收集和更新。数据主要来源于用户信息，用户可以是客户个人和社团群体两部分。客户个人输入主要指推荐系统的用户为了获得推荐而对一些项目进行评价，以表达自己的偏好。社团群体输入主要指集体形式的评价数据。

电子商务推荐系统的输入形式多种多样，主要包括以下几种方式：

1) 用户注册信息输入：用户在电子商务站点注册的时候，需要输入一些个人信息，这些信息可以是用户的年龄、性别、职业基本信息等，也可以是用户明确表达的喜好兴趣。这类信息是电子商务推荐系统收集到的关于特定用户的最初的信息。

2) 用户隐式浏览输入：将用户访问电子商务站点的浏览行为作为推荐系统的输入。这些由用户浏览行为提取出的信息并不需要用户刻意地对推荐系统进行配合，是在用户不察觉的情况下被存储的。用户当前正在浏览的产品、用户放入购物篮中的产品、用户的浏览路径等都可以作为隐式浏览输入信息。

3) 用户显式浏览输入：这也是将用户的浏览行为作为电子商务推荐系统的输入，但与隐式浏览输入不同的是，用户的显式浏览输入是有目的地向电子商务推荐系统提供自己的兴趣爱好。例如，电子商务系统提供一系列热门产品供用户选择，用户只选择浏览自己感兴趣的产品列表，电子商务系统根据用户的浏览行为向其提供个性化的推荐服务。

4) 关键字/产品属性输入：用户在搜索引擎中输入关键字作为推荐系统的输入，或将用户当前正在浏览的产品类别作为推荐系统的输入。这种类型的输入不同于用户随意的浏览行为，用户的输入目的就是在电子商务系统中搜索自己需要的产品。

5) 用户评分输入：将用户对产品的数值评分数据作为推荐系统的输入。电子商务推荐系统列出一系列产品让用户评分。用户的评分可以是一个数值，数值的大小表示用户对该产品的喜好程度；也可以是一个布尔值，0 代表不喜欢，1 代表喜欢。

6) 用户文本评价输入：用户对已经购买的产品或自己熟悉的产品以文本的形式进行个人评价。推荐系统本身并不能判断这些评价的好坏，但用户在浏览该产品时，可以通过其他用户对产品的文本评价信息来评判产品的好坏。

7) 编辑推荐输入：将领域专家对特定产品的评价作为推荐系统的输入。领域专家对产品的性能特点进行全面详细的介绍，用户通过专家的专业介绍，可以对自己并不熟悉的产品加深认识，从而决定是否购买该产品。

8) 用户购买历史输入：推荐系统将用户的购买历史作为隐式评分数据。一旦用户购买了特定产品，则认为用户喜欢该产品，推荐系统根据用户的购买历史记录产生相应推荐。但是用户购买了某件产品并不代表用户真正喜欢该产品，所以在精确的推荐系统中，用户可以对购买的产品进行重新评分，从而使推荐系统产生更精确的推荐。

(2) 输出模块 主要负责把推荐系统产生的推荐集输出给用户。电子商务推荐系统的输出主要包括以下几种方式：

1) 相关产品输出：推荐系统根据用户表现出来的行为特征或电子商务系统的销售情况向用户产生产品推荐，这种方式是电子商务推荐系统中最为普遍的一种输出。相关产品输出可以基于简单的销售排行向用户推荐热门产品，也可以基于对用户行为特征的深入分析，发现用户的购买模式，从而产生个性化的推荐。

2) 个体文本评价输出：电子商务推荐系统向目标用户提供其他用户对产品的文本评价信息。个体文本评价一般是非个性化的，对单个产品而言，所有用户得到的个体文

本评价都是相同的。

3) 个体评分输出：电子商务推荐系统向目标用户提供其他用户对产品的数值评分信息。个体评分输出没有大量的文本描述信息，因而更加简洁明了。个体评分输出比较适合于个体数值评分数据比较少的场合。

4) 平均数值评分输出：电子商务推荐系统向用户提供其他用户对产品的数值评分的平均值。这种输出形式具有简洁明了的优点，用户可以立即获得对该产品的总体评价。

5) 电子邮件输出：电子商务推荐系统通过电子邮件的形式向用户提供产品的最新信息。这种输出形式可以吸引用户再次访问电子商务站点，从而达到吸引用户关注度，防止用户流失的目的。

(3) 推荐方法模块是推荐系统的核心部分，负责由输入如何得到输出，决定着推荐系统的性能优劣。推荐方法模块以推荐技术和算法为技术支撑，具体的推荐技术将在2.2节详细介绍。

2.1.2 电子商务推荐系统的作用

电子商务推荐系统[3]就是利用统计和知识发现技术来解决与目标用户交互时提供商品推荐问题的系统。它在电子商务系统中模拟销售人员向客户提供商品信息和建议，帮助用户决定购买何种商品，向用户推荐商品，完成购买的过程。推荐系统推荐何种商品是根据电子商务网站上整体商品的购买情况、所有用户的购买历史记录等进行分析产生的。它根据分析得到的各个用户的兴趣爱好，分别推荐用户爱好的商品，因此也称为个性化推荐系统(Personalize Recommendation System)。

电子商务推荐系统的作用主要表现在以下几个方面：

1) 将更多的电子商务网站的浏览者转变为商品的购买者。电子商务网站的访问者往往没有购买欲望，通过预测用户的购买行为，主动为用户提供他们可能感兴趣的商品信息，从而促成交易。

2) 提高了用户对电子商务网站的忠诚度，与传统的商务模式相比，电子商务系统使得用户拥有越来越多的选择，用户更换商家极其方便，只需要点击一两次鼠标就可以在不同的电子商务系统之间跳转。推荐系统分析用户的购买习惯，根据用户需求向用户提供有价值的商品推荐。如果推荐系统的推荐质量很高，那么用户会对该推荐系统产生依赖。因此，电子商务推荐系统不仅能够为用户提供个性化的推荐服务，而且能与用户建立长期稳定的关系，从而有效保留客户，提高客户的忠诚度，防止客户流失。

3) 提高电子商务网站的交叉销售能力，电子商务推荐系统在用户购买过程中向用户提供其他有价值的商品推荐，用户能够从系统提供的推荐列表中购买自己确实需要但在购买过程中没有想到的商品，从而有效提高电子商务系统的交叉销售，为电子商务企

业赢得了更多的发展机会。例如用户购买了笔记本，网站为用户推荐笔记本锁。

目前，推荐系统已广泛运用到各行业中，推荐项目包括书籍、音像、网页、文章、新闻等。在日趋激烈的竞争环境下，个性化推荐系统能有效的保留客户，提高电子商务系统的服务能力，成功的推荐系统会带来巨大的效益。目前，几乎所有大型的电子商务系统都不同程度的使用了各种形式的推荐系统，如互联网上最大的书店—Amazon.com，互联网上最大的商店—CDNow.com，互联网上最大访问量之一的电影网站—MovieFinder.com 等。

2.1.3 电子商务推荐系统与个性化服务

所谓个性化服务，就是根据每个用户的不同喜好，为他们提供不同的服务，例如根据用户喜爱的页面，确定当前页面的下一级连接。网站在为用户提供个性化服务的同时，根据对用户兴趣的累积分析不断调整自己来适应用户兴趣的变化，使得每个用户都有是该站点唯一用户的感觉[4,5]。电子商务推荐系统使得电子商务系统主动适应每一个用户的特定需求，为每一个用户创建一个适应该用户的电子商店，从而为每一个用户提供完全不同的个性化购物体验，因此属于 Web 站点个性化服务的范畴。

不同电子商务推荐系统的个性化程度各不相同，根据电子商务推荐系统的个性化程度，可以将电子商务推荐系统分为如下三类[6]：

(1) 非个性化推荐系统

对每个用户产生的推荐都是相同的。这种推荐系统可以基于站点工作人员的手工推荐，或者基于统计分析技术等。我们经常见到的一些站点的销售排行、站长推荐、客户评论等，都属于非个性化电子商务推荐系统。

(2) 半个性化推荐系统

根据用户当前的行为产生相应的推荐。这种推荐系统根据用户当前的浏览行为或用户当前的购物篮信息产生推荐结果，一般使用关联规则等技术，不同用户得到的推荐结果各不相同。半个性化推荐系统的个性化程度比非个性化推荐系统要高。

(3) 完全个性化推荐系统

推荐系统保存用户的各种历史信息，如历史浏览信息、历史数值评分信息、用户注册信息等。根据这些历史信息，结合用户当前的行为，以及其他用户的历史信息，为用户产生完全个性化的推荐服务。这种推荐系统一般只能对注册用户提供服务，个性化程度最高。

2.2 电子商务推荐系统中推荐技术

为了使系统产生精确的推荐，保证推荐系统是实时性要求，电子商务推荐系统中目前已使用的技术主要有[7]：协同过滤(Collaborative Filtering)、关联规则(Association

Rules)[3]、聚类(Clustering)[8,9]、贝叶斯网络(Bayesian Network)、Horting 图(Horting Graph)[10]等技术。

2.2.1 协同过滤

协同过滤推荐(collaborative filtering recommendation)是目前研究最多的个性化推荐技术。一般采用最近邻技术，早期的协同过滤利用用户的历史信息计算他们之间的距离，再利用目标用户的最近邻居对商品评价的加权平均值来预测他对目标项的喜好程度，系统根据这一喜好程度来对目标用户进行推荐。而近些年，研究者[11]也有计算等项目间相似性，通过用户对相关项目的评分预测用户对未评分项目的评分。协同过滤最大优点是对推荐对象没有特殊的要求，能处理非结构化的复杂对象，如音乐、电影等。

目前有许多网站采用了该技术的推荐系统如表 2.1，此外由微软研究院开发的协同过滤工具已被集成在微软的 Commerce Server 产品中，并被许多站点使用。

表 2.1 采用协同过滤技术的网站

Table 2.1 Sites using collaborative filtering

系统名称	类型	网址
CDNow.com	CD 唱片	http://www.cdnow.com
GroupLens	Usenet 新闻	http://www.grouplens.org/
CoFE	电影	http://eecs.oregonstate.edu/iis/CoFE/
MoRec	电影	http://www.ug.boc.bilkent.edu.tr/~hakana/
MovieFinder.com	电影	http://movies.eonline.com/
MovieLens	电影	http://movielens.umn.edu/
Reel.com	电影	http://www.reel.com
Amazon.com	图书	http://www.amazon.com
Barnes&Noble	图书	http://www.barnesnoble.com/
Internet Watcher	网页	http://www.internetwatcher.com
SELEC	网页	http://www.dsv.su.se/~jpalme/select/
Jester	笑话幽默	http://shadow.ieor.berkeley.edu/humor/
Yenta	寻友	http://foner.www.media.mit.edu/people/foner/Yenta/
L.A.U.C.H.	音乐	http://www.lauch.com/
RACOFI	音乐	http://racofi.elg.ca/index.html

有关协同过滤的进一步介绍，见 2.3 节。

2.2.2 关联规则

关联规则技术在零售业得到了广泛的应用，关联规则挖掘可以发现不同商品在销售过程中的相关性。关联规则挖掘，就是发现数据集中项集之间有趣的关联或者相互联系，它是数据挖掘领域的一个重要分支，关联规则挖掘可以发现不同商品在销售过程中的相关性。基于关联规则的推荐（Association Rule-based Recommendation）[12]是以关联规则

为基础，把已购商品作为规则头，规则体为推荐对象。根据关联规则寻找相关项目，并对项目排序产生推荐[13]。

所谓关联规则，即在一个交易数据库中统计购买了商品集 X 的交易中有多大比例的交易同时购买了商品集 Y，得到的关联规则表示为 X=>Y[s%,c%]，s 表示关联规则的支持度，c 表示关联规则的置信度，其直观的意义就是用户在购买某些商品的时候有多大倾向去购买另外一些商品。比如购买牛奶的同时很多人会同时购买面包。即把已购商品作为规则头，推荐对象为规则体。简单的关联规则推荐算法过程如下：

(1) 使用关联规则发现算法，找出电子商务系统中满足最小支持度和最小置信度的关联规则 R。关联规则的发现算法很多，如 Apriori，AprioriTid，DHP，FP-tree 等。

(2) 在 R 中找出被目标用户支持的关联规则 R1，即规则左边的项目集是被目标用户购买过的项目集。

(3) 找出被关联规则 R1 所预测且没有被目标用户购买的所有商品 P。

(4) 根据 P 中商品在关联规则 R1 中的置信度排序，挑选前 N 个商品作为算法输出。如果某商品被多个规则预测，则取置信度最大者作为排序依据。

算法的第一步关联规则的发现最为关键且最耗时，是算法的瓶颈，但可以离线进行，因此可保证有效地推荐系统的实时性要求。其次，商品名称的同义性问题也是关联规则的一个难点。

2.2.3 聚类

聚类就是将数据对象归类，分为多个簇，这些对象与同一个簇中的对象彼此相似，而与其他簇中的对象相异[14]。聚类方法常用于协同过滤系统中。聚类法将有相似爱好的用户聚成组，完成聚类分析后，对当前用户的推荐可以通过和当前用户同类的其他用户的选择或观点进行平均化处理来得到。聚类方法推荐的结果通常比其它方法个性化要差些，在一些情况下，聚类结果比最近邻算法的精确度要低。但是，聚类一旦完成，最终效果可能会很好，因为这时要分析的组数要小很多。聚类方法可以用在最近邻法中作为“第一步”来缩小候选集。虽然将总体分为多个类可能会影响精确度或者推荐结果，但是，聚类法是值得在精确度与效率之间进行权衡的。由于聚类过程可以离线进行，所以在线的推荐算法产生推荐的速度比较快。Li 等人[15]采用基于项目的最近邻方法，先用 K-means 算法将项目进行聚类，选择活动用户已打分并且与目标项目在同一聚类中的项目，即将邻居限制在与目标项目在同一聚类的项目中，分别计算这些项目与目标项目之间的相似度，然后根据最近邻法求出活动用户对目标项目的评分预测值。Rectree[16,17]与这种方法类似，只是它是基于用户的方法，先将用户进行聚类，选择与活动用户在同

—聚类的那些用户作为潜在的邻居，然后根据这些用户与活动用户之间的相似度选择邻居。

2.2.4 贝叶斯(Beyesian)网络

贝叶斯网络(Bayesian Networks)是一种对不确定知识进行表达和推理的拓扑结构[18]，是人工智能、概率理论、图论、决策理论相结合的产物，它借助于图的直观表示和变换，清楚的表示变量之间的依存关系。在贝叶斯网络中，每一个节点表示一个变量，即一个事件；各变量之间的弧表示事件发生的直接因果关系。由于贝叶斯网络模仿了人的推理机制，所以它能很好的表达知识的不确定性。并且能够实现快速推理，因而在人工智能、目标识别、决策评估和信息融合等领域中得到广泛的应用[19]。

推荐系统中应用的贝叶斯网络技术是利用用户历史信息创建相应的模型，其中模型用决策树表示，节点和边表示用户信息。模型的建立可以离线进行，因为建模时间比较长，一般需要数小时或数天，而由此得到的模型可以非常小，对模型的使用非常快。但随着用户的不断增多以及用户兴趣爱好的变化，即数据集的变化，贝叶斯网络的学习过程也要重新进行，因此这种方法适合用户的兴趣爱好变化比较慢的场合。

2.2.5 Horting 图

电子商务推荐系统中的 Horting 图技术是一种基于图的方法[10]，节点代表用户，边代表两个用户之间的相似度。在图中寻找近邻节点，然后综合近邻节点的观点形成最后的推荐。Horting 图技术可以跳过中间节点寻找最近邻居，考虑了节点之间的传递相似关系。因此推荐精度优于最近邻协同过滤技术。

2.3 协同过滤介绍

在电子商务个性化推荐系统的各种算法中，协同过滤算法是应用最为成功的一类算法，在国内外尤其是国外各个大型电子商务网站的推荐系统中都得到了不同程度的应用。因而协同过滤推荐技术是个性化推荐领域重点研究的课题。

协同过滤，其原始基本思想是基于其他类似用户的兴趣来产生针对目标用户的兴趣推荐或预测。这个基本思想和现在颇为流行的“口碑传播(word-of-mouth)”有点儿类似。随着后来研究工作者的对协同过滤的不断深入研究，Sarwar[11]等人提出的基于项目的协同过滤推荐算法，这一思想又得到进一步扩展，通过收集其他相似用户或相似项目的信息来预测当前用户的兴趣。

2.3.1 协同过滤系统简单描述

协同过滤系统可以由输入，协同过滤引擎以及输出三个部分组成[20]，即用户输入评

价信息，协同过滤引擎根据用户输入的信息产生预测或推荐，输出结果这三个步骤。一般来说，预测引擎对用户来说是个“黑盒”，用户是不知道给他的推荐结果是怎样得到的。

第一步为用户输入他对一些项目的评价。正如上一章所提到的，推荐系统的输入的用户评价可以是显示和隐式，但是隐式评价对数据分析难度较大，准确性、有待进一步提高，所以协同过滤领域研究主要以显示评价为主。这里我们假定某一个推荐系统有 m 个用户和 n 个项目，使用 $m \times n$ 矩阵 R 来表示用户评分信息，矩阵中的每一个评分 $r_{i,j}$ 表示用户 i 对项目 j 的评分。

$$r_{i,j}=\begin{cases}\text{实际评分，} & \text{如果用户}i\text{对项目}j\text{投票}\\ 0, & \text{如果用户}i\text{对项目}j\text{没有投票}\end{cases} \tag{2.1}$$

对于推荐要预测的用户本文称作目标用户，在图 2.1 中的目标用户是 i，要预测的是其对项目 j 评分。

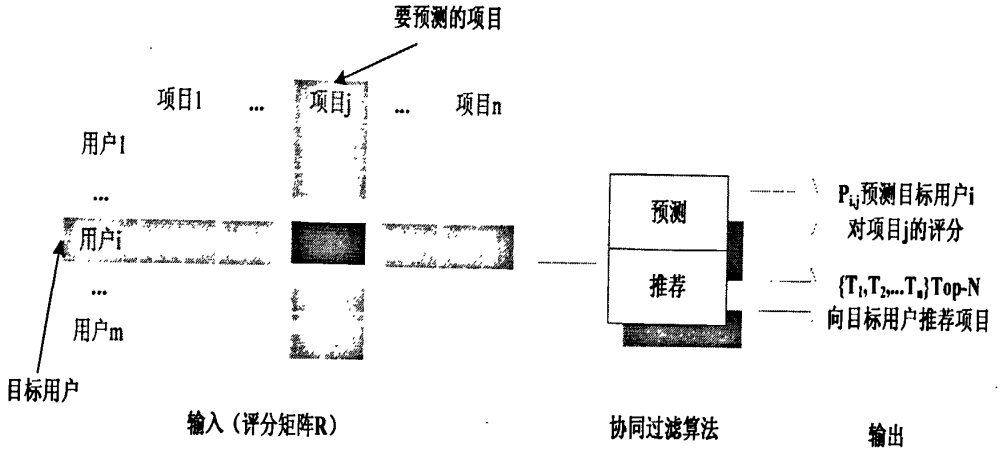

图 2.1 协同过滤过程

Fig.2.1 The collaborative filtering process

第二步就是收集所有评分值，从现有信息资料中归纳出用户兴趣模板，然后根据这些信息利用上文我们提到的那些协同过滤技术来从海量信息中过滤掉用户不需要的，并为目标用户返回一个的有序序列，或者是当一个目标用户提供给预测算法一个项目的列表后，预测算法返回这个列表中各个项目的预测分值。

第三步就是要输出预测的结果。输出的预测结果主要有两种形式，一种是推荐，另外一种是预测。推荐是向目标用户提供一份用户可能感兴趣的项目的列表，典型的如 TopN，根据客户的喜好向客户推荐最可能吸引客户的 N 件产品。预测就是系统对给目标用户对特定项目的评分估计一个值。

2.3.2 协同过滤技术的分类

研究人员已经提出了大量的协同过滤算法，主要有两种分类方法。

Breese[21]等人根据协同过滤所采用的算法，将其分为两种主要的类别，即基于内存的(memory-based)与基于模型的(model-based)两种。

(1) 基于内存的协同过滤：基于内存的协同过滤是最流行的预测方法而且在商业协同过滤系统中广泛被采用。这些算法根据用户项目评分矩阵，利用统计的方法得到具有相似兴趣爱好（倾向于购买相似的产品或对于同一项目评分相似）的邻居用户集或与预测项目相似（同一用户项目的评分相似）的邻居项目集，再基于邻居进行计算，使用不同算法来结合邻居集的评分产生对于目标用户的预测或 top-N 项目推荐。这些技术又叫最近邻方法(nearest neighbors)，在实践中是比较流行和广泛。

(2) 基于模型的协同过滤：在基于模型的协同过滤中，训练集被用来训练一个预先定义的模型。其主要是将使用者历史记录，通过统计方法或机器学习方法来建构出使用者偏好模型，进而利用此偏好模型来产生推荐，目前所使用的方法包括关联规则法(Association Rule)、贝叶斯网络(Bayesian network)、回归分析(Regression analysis)等等。

依据协同过滤技术所使用的事物之间的关联性，将其区分为基于用户的与基于项目的协同过滤技术：

(1) 基于用户的协同过滤：其核心概念是假设人与人之间的行为具有某种程度的相似性，即购买行为类似的顾客，他们会购买相类似的产品。GroupLens 即属于此类型的系统；

(2) 基于项目的协同过滤：其主要假设是项目与项目间具有某种程度的关联，即顾客在购买时，其所购买的产品通常具有关联性，如顾客在购买电子游戏机时，通常会购买电池及游戏卡。目前应用较多的是基于邻居用户的协同推荐算法。

2.3.3 现有的协同过滤推荐系统

Typestry[22]是最早提出来的基于协同过滤的推荐系统,但需要目标用户明确指出与自己行为比较类似的其他用户。GroupLens[23]是基于用户评分的自动化协同过滤推荐系统,用于推荐影片和新闻。Ringo 推荐系统[24]和 Video 推荐系统[25]通过电子邮件的方式分别推荐音乐和影片。

这里我们主要介绍一下 GroupLens，GroupLens 是一个应用于 Usenet 新闻的协作过滤系统，它的目标是让用户一起协作，从大量的 Usenet 新闻中发现他们感兴趣的内容。系统分为两部分：客户端和服务器端.客户端是一个新闻阅读器 NewsReader，服务器端提供协作过滤。NewsReader 一般连接到本地 NNTP 服务器，同时也连接到 GroupLens 服务器共享过滤信息，只要用户下载一篇档，NewsReader 都会向 GroupLens 服务器发

送消息请求对该文档内容的预报，也就是其他用户对该文档的评价。此外，用户也可以评价文档，NewsReader 会将该用户评价发送到 GroupLens 服务器上进行处理，以提供给其他用户浏览，GroupLens 会利用这些信息调整该用户和其他用户的相关性。

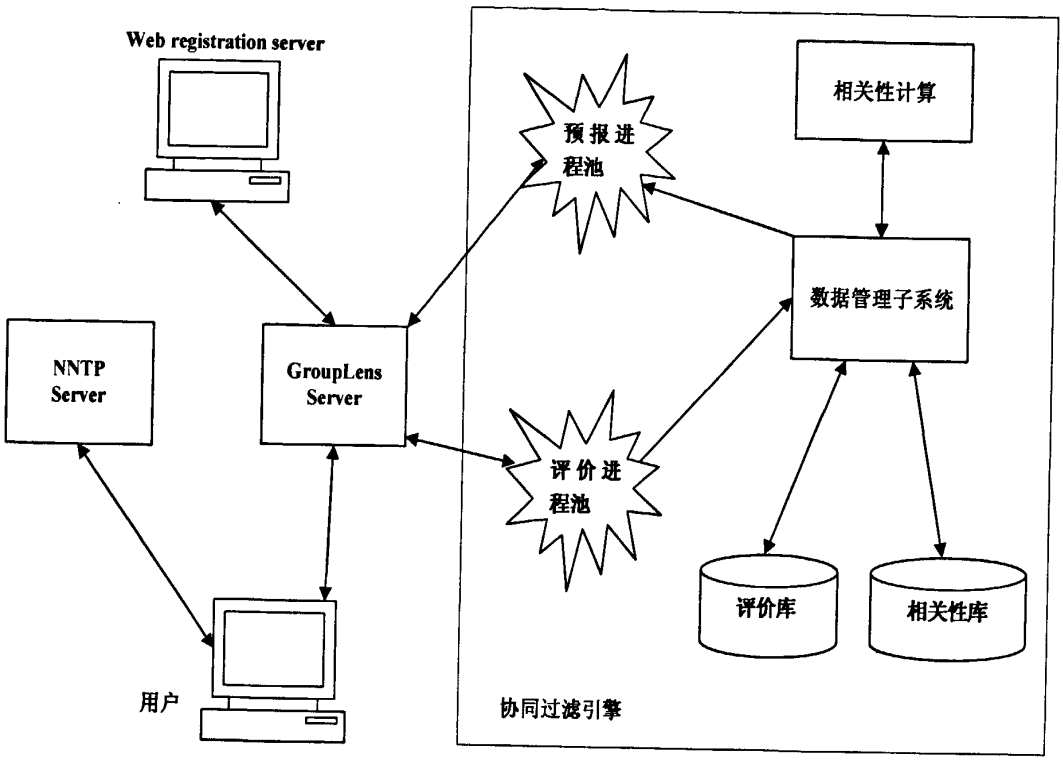

图 2.2 GroupLens 体系结构

Fig. 2.2 Architecture of GroupLens

GroupLens 服务器端分为请求代理和过滤引擎。请求代理将用户的请求分派到合适的进程来处理。过滤引擎分为四部分（如图 2.2 所示）：预报模块、评价模块、相关性计算模块、数据管理模块。预报进程为客户提供精确的文档内容预报，文档的预报处理需要两类信息：用户间的相似性和用户对文档的评价；评价进程接收用户的评价并将其安全存储；用户相关性计算程序计算用户间的相似性；数据管理子系统管理用户的评价和用户相似性等数据。

2.4 协同过滤存在问题以及现有解决方法

2.4.1 协同过滤在应用中存在的问题

尽管协同过滤在电子商务推荐系统中的应用获得了较大的成功，但随着站点结构的复杂化，商品信息量和用户与日俱增，基于协同过滤的推荐系统的发展面临着两个主要挑战:

(1) 提高协同过滤算法的可扩展性

协同过滤算法能够容易地为几千名用户提供较好的推荐，但是随着电子商务网站的

发展，用户数已经达到成百上千万，这就一方面需要提高响应时间的要求，能够为用户实时地进行推荐另一方面还应考虑到存储空间的要求，尽量减少推荐系统运行的负担。

(2) 提高推荐信息的质量

用户需要值得信任的推荐系统来帮助他找到自己喜欢的产品。假如推荐系统老是推荐用户不喜欢的商品，或者用户相信推荐购买了商品，而后发现自己并不喜欢，用户对推荐系统推荐结果的信任度降低，同时将不愿再次使用该推荐系统。

从一定意义上讲，推荐系统面临的这两个挑战之间存在着矛盾，系统要提高算法的可扩展性及响应时间，在质量上必然会有所损失。因此，如何协调好这两方面的要求，使推荐系统不仅有用而且实用，是实现协同过滤技术需要考虑的重要因素。以下我们将叙述协同过滤技术的实现中存在的具体问题。

2.4.1.1 稀疏性问题

稀疏问题(sparsity)是推荐技术中的重要问题之一[26]。协同过滤技术的基础是基于用户的历史信息，包括目标用户和其它用户，这些信息可通过用户一项矩阵来表示。实际中，许多电子商务推荐系统需要对大量项目集合进行评价[11]（例如，Amazon.com 推荐书籍和 CDnow.com 推荐音乐专辑）。在这些系统中，一般用户购买商品的总量最多占网站总商品量的 1%左右，因此造成了用户评分矩阵非常稀疏。在这种数据量大而且又稀疏的情况下，一方面难以找到最近邻居用户集，相应地，推荐系统可能不能够为某个特定用户作任何项目推荐，另一方面进行相似性计算的耗费也会很大。

同时，由于数据非常稀疏，在形成目标用户的最近邻居用户集时，往往会造成信息的丢失[27]，从而导致推荐效果的降低。例如，邻居用户关系传递性的丢失。用户 A 与用户 B 相关程度很高，用户 B 与用户 C 相关程度也很高，但由于用户 A 与用户 C 很少对共同的产品进行评价，而认为两者关联程度较低，由于数据的稀疏性，丢失了用户 A 与用户 C 之间潜在的关联。

2.4.1.2 冷开始问题

又称第一评价问题(first-rater)，或新项问题(new-item)[28][29]，从一定角度可以看成是稀疏问题的极端情况。因为传统的协同过滤推荐是基于邻居用户资料得到目标用户的推荐，在一个新的项首次出现的时候，因为没有用户对它作过评价，因此单纯的协同过滤无法对其进行预测评分和推荐。而且，由于新项出现早期，用户评价较少，推荐的准确性也比较差。相似的，推荐系统对于新用户的推荐效果也很差。冷开始问题的极端的案例是：当一个协同过滤推荐系统刚开始运行的时候，每个用户在每个项上都面临冷开始问题。

2.4.1.3 可扩展问题

前文我们提到协同过滤算法可以分为基于内存和基于模型两类。基于内存协同过滤

算法能及时利用最新的信息为用户产生相对准确的用户兴趣度预测或进行推荐，但是目前大多数电子商务系统的用户都很多，商品信息更多，而一般的协同过滤算法却不能适应这种膨胀，性能也越来越差。这就是协同过滤算法的扩展性问题。虽然与基于模型的算法相比，基于内存的协同过滤算法节约了为建立模型而花费的训练时间，但是用于识别“最近邻居”算法的计算量随着用户和项的增加而大大增加，对于上百万的数目，通常的算法会遇到严重的扩展性瓶颈问题。该问题解决不好，直接影响着基于协同过滤技术的推荐系统实时向用户提供推荐问题的解决，而推荐系统的实时性越好，精确度越高，该系统才会被用户所接受。

基于模型的协同过滤算法虽然可以在一定程度上解决算法的可扩展性问题，但是该类算法往往比较适于用户的兴趣爱好比较稳定的情况，因为它要考虑用户模型的学习过程以及模型的更新过程，对于最新信息的利用比基于内存算法要差些。

2.4.2 现有的解决办法

2.4.2.1 SVD 降维

奇异值分解(Singular Value Decomposition,SVD)，是一种矩阵分解技术，它是一种有效的代数特征提取方法，深刻揭露了矩阵的内部结构[30,31]。目前，奇异值分解在信息检索方面的应用主要是隐含语义检索(Latent Semantic Indexing,LSI)。为了较好地解决协同过滤在推荐系统实现中存在的数据稀疏、同义词等问题，文献[29] 使用奇异值分解方法将用户评分分解为不同的特征及这些特征对应的重要程度，这种方法利用了用户与项目之间潜在的关系，用初始评价矩阵的奇异值分解去抽取一些本质的特征。通过奇异值分解减少项目空间的维数，使得用户在降维后的项目空间上对每一个项目均有评分，实验结果表明，这种方法可以有效地解决同义词问题，能提高推荐系统的可扩展性，应用于协同过滤时对于稀疏的评分矩阵效果比较好，显著地提高推荐系统的伸缩能力。

但是在使用 SVD 技术之前，需要对用户项矩阵进行规范化，例如将矩阵中评估值为 0 的项用相关列的平均值代替，即项的平均评估值，接着将矩阵每行规范化为相同长度。选择不同数量项的用户对相似度计算结果的影响不同，容易造成偏差，规范化为相同长度后，降低了选择项目数较多的用户对相似度计算结果的影响。也正因为如此，规范化会使矩阵不再能够完全真实地反映用户的信息，导致信息损失，从而也就降低了推荐的质量。这使得该方法在项目空间维数很高的情况下，难以保证推荐效果[32]。

矩阵的奇异值分解计算量通常比较大，但是可以离线进行。

2.4.2.2 特征加权

在协同过滤算法中用户之间的相似性可以通过计算皮尔森相关系数或向量相似性等方法来度量，但计算公式中对两个用户评价过的所有项的处理上是完全相同的，即处

于同等地位没有重要与非重要之分，这可能会对预测结果的准确性上有一定的影响。因此，使用一些加权的方法来控制不同项用户信息的描述项或特征项对用户兴趣度预测的影响[21]，减小甚至消除某些项产生的消极影响，提高与目标项紧密相关项的影响，这样在一定程度上会提高推荐结果的质量。下面介绍一下常用的几种方法[27]：

1．逆用户频率(Inverse User Frequency)

在信息检索的向量相似性的应用中使用倒排文件频率有效地改善了单纯词频的使用，它的主要思想是减小在文档中经常出现，但对文档主体识别不是非常起作用词语的权重对出现频率低，但是对文档主题识别非常有用的词语赋予较高的权重。在文献[21]中将类似的这种技术运用到协同过滤中并称之为逆用户频率，其主要思想是对于那些有许多用户评价过的项不如被少数用户评价过的项更。由此，倒排用户频率公式如下：

$$\omega_j = \log \frac{n}{n_j} \tag{2.2}$$

n_j 为所有对项 j 进行过评价的用户的总数，n 为数据库中用户的总数，如果所有的用户都对项 j 进行了评价，则 ω_j 的值为 0。当然，如果对于所有的项进行评价过的用户数目都相同，那么使用该权重也就没有意义了。

2．熵(Entropy)

熵用于衡量随机变量的不确定性。在协同过滤算法中用户对某一项或产品评价的分布非常重要，假设如果所有的用户对某产品的评价值都较高，那么计算用户之间的相似性没有意义，因为它说明不了用户之间的区别。但是，如果用户对产品评价在整个范围值内分布分散，用户评价差异较明显，对预测目标用户对该产品的偏爱程度就比较有意义。基于以上的想法，文献[34]中提出了基于熵的权重方法，公式如下：

$$\omega_j = \frac{H_j}{H_{j,\max}} \qquad \text{其中 } H_j = -\sum_i p_{i,j} \cdot \log_2 p_{i,j} \tag{2.3}$$

公式中 H_j 表示产品 j 的熵，$p_{i,j}$ 表示评价值 i 在对产品 j 的评价中出现的概率，$H_{j,\max}$ 表示假设对产品 j 所有类型的评价值概率分布相同的情况下的最大熵，使用它是为了减小用户对不同产品进行评价时，由于评价值不同、分散而产生的影响。这样 ω_j 值越大表明用户对产品 j 比较偏爱，该产品对预测的影响较大。然而，如果不同产品间的熵相差不大，同样使用该方法也就没有了意义。例如在电影推荐中人们对每一部电影的爱好程度都会有很大的不同，这样对于许多电影熵值可能会为 1，这样基于熵的权重方法失去了作用。

3．互信息(Mutual Information)

以上两种方法均是从产品或项的自身特点出发来考虑的，并没有涉及到其它项与目

标项之间的关系，因为如果项j 对目标项的预测非常重要，可以赋予它较高的权重，而与目标项相关性程度低的项通过权重降低影响，从而提高推荐系统结果的质量。文献[34]提出使用互信息来衡量不同项之间依赖程度，并以此作为特征权重：

$$\omega_i = I(V_i;V_t)$$

$$I(V_i;V_t) = H(V_j) + H(V_t) - H(V_j,V_t) \tag{2.4}$$

V_j 与V_t 分别表示对项j 和项 t 的评价值，$H(V_j,V_t)$ 是两项的联合熵。由于不是所有的用户均对两项作了评价，因此计算只在对两项均进行了评价的用户中进行。如果在训练数据集中有 n 个这样的用户 m 个项，计算所有项之间的互信息的复杂度为 $O(nm^2)$，而 n 往往远大于 m。

特征加权的方法主要用来提高推荐质量。

2.4.2.3 用户的筛选

基于内存的协同过滤算法是基于这样的假设具有相似兴趣的用户会对相同的项目感兴趣。然而，在实际中这样的假设并不总是成立的。因此，为了提高系统结果的准确度，一方面可以通过给不同项赋予不同的权重，另一方面可以在最近邻居用户的选择上作一定的改进。

在前面我们已经提到，计算用户间的相似性之后，一般按照相似性值的大小选取最近邻居用户。如果有更好的方法选取这一集合中的数据，找到目标用户的“真正的”邻居，一方面提高预测的准确度，另一方面减少计算的复杂度。根据文献[35]的研究，下面介绍两种实现的方法：

1．选取具有新颖描述的用户

该方法的主要思想是对于相似性程度差别不大的多个用户，可以只保留其中的一部分，去除的用户对推荐结果的影响可以忽略不计，但由于计算量的减少反而加快了系统运行的速度。描述如算法 2.1：

该算法的优点在于以下几点：

1) 充分考虑到了邻居用户的评价值相互之间不一致时，用户评价值变化比较明显的那部分用户。

2) 避免了由于多数用户评价值过于集中造成的误差，因为由于数量多这些值往往会比其它最近邻居用户特别是关键的最近邻居用户产生更大的影响，从而导致偏差。

3)对于新的用户偏好模式能及时根据判断加入到最近邻居用户集中。

实验证明该算法能有效减少每一项进行预测计算的用户数，提高了预测速度和准确度。但是算法也存在不足，如由于过多考虑到了评价值比较例外的用户，往往会把一些用户作为最近邻居用户加入，这样的用户对目标项的评价值，即使通过该用户本身对其

算法 2.1 用户的筛选

输入：整个训练数据集 T；初始用于预测的用户集中的用户数 Initial_Size

输出：每一项的最近邻居用户集T_i'

步骤：

1:对 T 中的每一项 i

2:如果(所有评价过项 i 的用户集T_i中的用户数>>Initial_Size)

3:从T_i中随机选 Initial_Size 数目的用户形成初始的用户集T_i'

4:对于每一个T_i中的剩余用户 u

5:如果(u 的评价值不能通过预测公式在T_i' 范围内正确的预测到)

6:那么将 u 加入T_i'

这样，对于每一项 i 都有了其相应的缩小了的评价过该项的用户集 T_i'。

它项的评价值也无法进行解释，如同数据噪音，导致了预测的失败。另一个不足就是从算法中可以看到由于对T_i中的每一用户都要计算在当前最近邻居用户集T_i'下的预测值，当数据集很大时，这种耗费可以说会很大。因此该方法往往与其他方法结合，首先进行了一定程度的用户过滤以后，再考虑使用该方法进行进一步的用户过滤。

2. 选取具有合理描述的用户

该算法的主要集中解决的问题是对任一用户，能否通过他在数据集中的数据较好地描述出来。前面一节可知互信息表示项与项的相关性，综合考虑用户描述项与目标项的相关性来进行用户的选择，但实际上并不是用户的描述项越多，用户与目标项之间的合理度(rationality)会越高，因此在文献[35]中使用合理性强度(the strength of rationality of an instance u)来进行用户的选择，公式如 2.5。

$$R_{u,i}^{strength}=\frac{1}{|F(u,i)|}R_{u,i}=\frac{1}{|F(u,i)|}\sum_{j\in F(u,i)}I(V_i;V_j) \tag{2.5}$$

$F(u,i)$ 表示用户 u 对项 i 以外的其它项的评价值集合，$F(u,i)$ 成为用户 u 的描述项集。$I(V_i;V_j)$ 为项目 i 和j 之间的互信息。

算法是这样实现的：首先对项与项之间评价值的互信息进行计算，对每一目标项 i，使用公式(2.5)计算评价 i 的所有用户的合理性强度，按照从高到低排序，然后根据一定的比例 r 选取用户(该比例会影响预测结果的准确性及运行的效率)，作为预测时衡量相

似性大小选取最近邻居用户的基础，最后进行预测计算。

实验证明该方法提高了推荐结果的质量，而且最近邻居用户的数量有明显的减少。在训练阶段(预测计算之前的操作)的代价要比前一种方法低，假设在训练集中有 n 个用户，m 个项，则在训练阶段的计算复杂度为 $O(nm^2)+O(nm)+O(nlogn)$，实际上 m 往往比较稳定，随着用户的不断增加，n 的动态变化较大，结果往往会是 n 远远大于 m，而且该算法根据比例 r 值的不同，效率和准确度会有所不同，r 最佳值的获取是这一算法需要考虑的问题。

该类方法的主要目的在于解决协同过滤算法的扩展性问题。

2.5 小结

本章首先对电子商务推荐系统的相关知识进行了介绍，从整体上对电子商务推荐系统有了认识和了解，着重介绍了电子商务推荐系统中主要使用的推荐技术。随着对推荐系统功能需求水平的不断提高，其实现技术也面临着严峻的挑战。本章重点介绍这一在电子商务推荐系统中应用比较成功的协同过滤技术，介绍了协同过滤技术的基本思想、实现过程、现有的协同过滤系统以及协同过滤技术的分类，并对协同过滤存在问题和相应解决方法进行了详细的说明与分析。

第三章 基于项目协同过滤的类型优化算法

3.1 传统基于项目协同过滤算法

基于用户的协同过滤系统运行的瓶颈是要在一个很大的用户群中找出合适的邻居，基于项目的协同过滤系统可以通过寻求项目之间的相似关系，而不是用户之间的相似关系来避免这个瓶颈问题。

基于项目的协同过滤(item-base CF)是通过用户－项目矩阵分析每个项目间的相似性，在此基础上得到被推荐的前N个项目。这个方法的根本思想是：一个用户将更会喜欢那些和他已经购买的项目相似的项目。

两个项目之间相似性计算是在用户评分矩阵的列之间展开，如图 3.1，其基本思想是先分离出所有已经对这两个项目进行了评分的用户，然后应用相似性计算决定这两项的相似性。

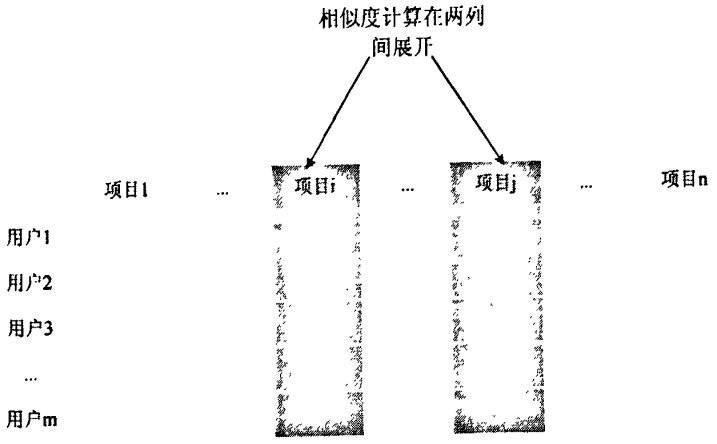

图 3.1 基于项目协同过滤相似度计算

Fig.3.1 Similarity computing of item-based CF

因为在典型的电子商务环境中，项目之间的关系相对来说比较稳定，所以利用项目之间的相似性，基于项目的协同过滤算法可以花费较少的在线计算时间来得到与基于用户的协同过滤系统准确性相近的预测结果，这种方法在某种程度上解决了基于用户的协同过滤系统中存在的可扩展性问题。

基于项目的协同过滤方法可以分为三个步骤：计算项目之间的相似程度，确定最近邻居集和产生推荐。其中一个至关重要的步骤是计算项之间的相似性，然后来选择最相似的项目，项目之间的相似性计算方法有很多种。下面我们将进行进一步说明。

3.1.1 相似度计算

如何度量项目与项目之间的相似程度？目前主要有余弦（cosine）相似性、修正余弦(adjusted cosine)相似性和相关(correlation)相似性。

(1) 余弦相似性：项目评分被看作为 m 维用户空间上的向量，如果用户对项目没有进行评分，则将用户对该项目的评分设为 0。设项目 i 和项目 j 在 m 维用户空间上的评分分别表示为向量 $\vec{i}$、$\vec{j}$，则项目 i 和项目 j 之间的相似性 $sim(i,j)$ 为：

$$sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|\cdot\|\vec{j}\|} \tag{3.1}$$

分子为两个项目向量的内积，分母为两个用户向量模的乘积。

(2) 修正余弦相似性：由于在余弦相似性度量方法中没有考虑不同用户的评分尺度问题，例如有些人常给高分，而有些人给的评分普遍较低。为了克服这一缺陷，修正余弦相似性方法中减去用户对项目的平均评分。设项目 i 和项目 j 共同评分的用户集合用 U_{ij} 表示，U_i 和 U_j 分别表示给项目 i 和项目 j 评分的用户集合，$\overline{R}_u$、$\overline{R}_v$、$\overline{R}_w$ 分别表示用户 u、v、w 已评分的项目的平均值,则项目 i 和项目 j 之间的相似性 $sim(ij)$ 为：

$$sim(i,j)=\frac{\sum_{u\in U_{ij}}(R_{u,i}-\overline{R}_u)(R_{u,j}-\overline{R}_u)}{\sqrt{\sum_{v\in U_i}(R_{v,i}-\overline{R}_v)^2}\sqrt{\sum_{w\in U_j}(R_{w,j}-\overline{R}_w)^2}} \tag{3.2}$$

(3) 相关相似性：设对项目 i 和项目 j 共同评分的用户集合用 U_{ij} 表示，$\overline{R}_i$、$\overline{R}_j$ 分别表示项目 i 和项目 j 所有对其评分的平均分，则项目 i 和项目 j 之间的相似性 $sim(i,j)$ 通过 Pearson 相关系数度量为:

$$sim(i,j)=\frac{\sum_{u\in U_{ij}}(R_{u,i}-\overline{R}_i)(R_{u,j}-\overline{R}_j)}{\sqrt{\sum_{u\in U_{ij}}(R_{u,i}-\overline{R}_i)^2}\sqrt{\sum_{u\in U_{ij}}(R_{u,j}-\overline{R}_j)^2}} \tag{3.3}$$

3.1.2 最近邻居

根据计算好的项目之间的相似性，对于目标项目 i，按照其与其他项目的之间的相似性从大到小排序，产生一个最近邻居候选集合 $C=\{i_1, i_2, ..., i_t\}$，i 不属于 C。再根据要预测某一用户 u 对目标项的评分，从中选择已被该用户评价了的项目。在预测不同的用户的评分时，项目的邻居并不一定都是一样的,最后得到针对要预测用户 u 对项目 i 的评分的最近邻居集合 N。

3.1.3 产生推荐

根据最近邻居集合 N 产生推荐，预测用户 u 对项目 i 的评分 $P_{u,i}$，对于基于项目协

同过滤，以下公式较为常用并取得较好效果：

$$P_{u,i}=\overline{R}_i+\frac{\sum_{j\in N} sim(i,j)\times(R_{u,j}-\overline{R}_j)}{\sum_{j\in N}|sim(i,j)|} \tag{3.4}$$

其中 $R_{u,j}$ 表示用户 u 对项目 j 的评分，$sim(i,j)$ 为项目 i 和 j 的相似度，$\overline{R}_i$、$\overline{R}_j$ 分别表示所有对项目 i 和项目 j 的评分的平均分。

若要推荐项目，可通过上述方法预测用户对所有未评分项目的评分，然后选择预测评分最高的前若干个项目作为推荐结果反馈给当前用户。

3.2 基于项目协同过滤的类型优化算法

3.2.1 问题的提出

在各种不同的个性化推荐系统中，都有一些项目自身的所包含的信息，以电影网站为例，电影本身信息都包含电影名、发行时间、类型、导演、编剧等信息。而网站就将电影按照动作片、喜剧片等类型进行归类，其实这样是为了方便用户更好地找到自己可能喜欢的影片，这也正好说明影片类型代表影片特征的重要性，这是左右用户的选择的重要因素。个性化推荐面向的就是用户，所以在这些信息中，项目分类信息对于个性化推荐是较为重要的，其对于个性化推荐的质量的影响是较为重大的。

在各个不同的推荐系统中，对所提供的项目都以大的类别来划分，可以表示为：

$$\begin{aligned}
&I=S_1\cup S_2\cup\cdots\cup S_k,\\
&S_1=\left\{i_{11},i_{12},\cdots,i_{1j_1}\right\},\\
&S_2=\left\{i_{21},i_{22},\cdots,i_{2j_2}\right\},\\
&\vdots\\
&S_k=\left\{i_{k1},i_{k2},\cdots,i_{kj_k}\right\}.
\end{aligned}$$

这里 I 表示所有项目组成的集合，S_i 表示第 i 类，用 $Num(X)$ 表示 X 集合的元素数目，则 $Num(I)\le j_1+j_2+\cdots+j_k$。

这样的划分，很常见。而属于同一类型的项目无疑一定程度上是存在某些共同之处的。而在传统计算项目相似性的方法中体现不出这种类型划分所包含的实际意义。例如，项目 i 和 j 项目的类型有交叉（有一些共同含有的类型），而 i 和 k 没有交叉（没有任何类型相同），但是如果项目 j 和 k 所得到的评分完全一样，那么根据传统相似度方法（参见上一节）计算出的项目相似性 $sim(i,j)=sim(i,k)$，而这样计算出的项目相似性就不是实际应该有的结果了。虽然我们所举的例子有些极端，但是说明了计算项目间的相似性单单只看用户对项目的评分是不够的。而且通常由于用户评分矩阵的稀疏，项目的相似性计算结果大多都比较小，而相似度之间的差异就更小，如果 $sim(i,j)$ 和 $sim(i,k)$ 的不相等

而是差异很小，而其实二者是差别应该是比较大的，这导致最终预测评分时没有了差别性，甚至可能产生负面影响。同样因为用户评分矩阵的稀疏性，有些项目只有几个用户评价，而恰恰这评价的几个用户同时评价了其中几个项目，而且评分相近，这只能说这几个用户都同时对这两项目感兴趣，并不能说明这些项目间的相似性真的是这样的，这里我们称之为用户不可信，即对项目的用户评价个数很小，导致仅依赖评分的项目相似性计算方法的可信度也大打折扣。

综上所述，传统的基于项目协同过滤推荐算法的相似度度量方法存在以下一些问题：

(1) 在计算用户相似性时，往往只利用用户评分矩阵进行相似度的计算，因而在用户评分数据极端稀疏的情况下，传统的相似性度量方法不能有效的计算项目之间的相似性；

(2) 没有充分利用项目类型信息，而项目类型信息是项目固有信息，对项目的相似性具有很高的参考价值。在用户评分数据极其稀疏的情况下，在协同过滤算法中充分利用项目类型信息显得很必要。

3.2.2 相关工作

近些年，随着协同过滤方面的研究工作的不断深入，陆续出现将项目自身信息引入基于项目协同过滤的研究成果，在文献[36]中利用项目信息中的类型信息，判断两个项目是否算是一个类，在计算项目间的相似性中引入该值。但其判定项目是否为一类的方法过于简单，因为一个项目往往属于多个类或者说含有多种类型，所以说其并没有充分利用项目类型信息，不是很准确。

3.2.3 项目类型矩阵

很多推荐系统都有关于项目类型的信息，这些项目信息可以直接获得，在基于项目协同过滤算法中如何利用项目类型信息呢？我们通过项目类型矩阵来表示项目类型信息，例如电影网站，MovieLens 站点(http://movielens.umn.edu)就将电影分为 19 个类型，例如：

飘(Gone with the Wind)这部电影含有战争(war)、爱情(romance)、戏剧(drama)三种类型。

项目类型矩阵的定义：设推荐系统有 n 个项目和 k 个类型，每个项目分别归于这 k 个类型中的一些，项目类型矩阵如表 3.1。

表 3.1 项目-类型矩阵

Table 3.1 Item-genre matrix

	类型 1	类型 2	类型 3	...	类型 k
项目 1		1	1	...	
项目 2			1	...	1
项目 3		1	1	...	1
...	...	...	...	...	...
项目 n	1		1	...	

其中 1 代表项目可以归于这个类型或者说项目含有这个类型。

我们可以使用 0 填补空白，代表该项目不含有这个类型。这样项目类型矩阵是一个二进制矩阵。

3.2.4 类型优化

从上文的分析我们可以看出，项目类型在相似性计算上有较大的影响。但是虽然不是那么显而易见，我们不能否认用户项目评分数据中隐含了项目类型信息的成分，所以我们只需要强调这一点而无须抹煞用户评分价值。因此，我们在传统基于项目协同过滤推荐算法仅基于用户评分数据的相似性度量的基础上引入项目类型矩阵参与项目相似性的计算。

我们提出的算法与 4.1 提到的传统的基于项目协同过滤推荐算法步骤基本一致，即相似度计算、确定最近邻居和产生推荐三步。二者主要差异在于相似性计算这一环，这里主要就这一步进行说明。

首先按照传统的相似性度量方法通过用户评分矩阵计算相似性，使用 4.1.2 提到的方法，具体方法选择，下文通过实验进行选择，这里我们将项目 i 和 j 的基于用户评分矩阵计算的相似性记作 $sim_R(i,j)$。

然后利用上一节中介绍的项目-类型矩阵，计算项目之间的类型相似性，由于是项目类型矩阵元素是二进制值，所以使用余弦相似性来计算，较为快捷和合理，记作 $sim_{genre}(i,j)$,如公式 3.5：

$$sim(i,j)_{genre} = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|\cdot\|\vec{j}\|} \tag{3.5}$$

这里 $\vec{i}$、$\vec{j}$ 分别为项目 i 和 j 在项目-类型矩阵对应的行向量，分别代表了项目 i 和项目 j 所含的类型。

我们采用线性结合的方式将传统项目间相似性度量结果和项目类型相似度结果向结合，共同作为两个项目间的相似性，形式如下：

$$sim(i,j)=(1-\lambda)\times sim_R(i,j)+\lambda\times sim_{item-genre}(i,j) \tag{3.6}$$

其中λ是一个在[0，1]范围内的参数，该参数允许我们设定项目或用户间相似性依赖于传统相似性和类型相似性的比重。同时也一定程度上平衡用户评分矩阵与项目类型矩阵的疏密性的差异。注意：$sim_R(i,j)$和$sim_{item-genre}(i,j)$虽然都是计算项目i和j之间的相似性，但是基于不同的数据计算的，前者是使用用户项目评分矩阵，而后者是使用项目类型矩阵。

这里我们进一步讨论一下参数λ的取值范围，即类型相似度与传统相似度之间所占的比重，正如我们在前文提到的那样，由于用户评分数据不止包含了用户的喜好和项目之间的关联，而类型信息这个项目本身信息也隐式地包含在其中。而推荐系统是以人为服务对象，因此我们所要的是要加强类型对项目相似性度量的贡献，因为其对于分析项目的相似性是很重要的，所以只需加重类型的比重，而不能完全抹杀传统项目相似性度量中其他因素的作用（如电影的演员、导演、出品国家等）。由此可知，传统项目相似度的比重应该大于类型相似度的比重，即(1-λ)应该大于λ，那么λ应该在[0,0.5)范围中，而且λ不能为 0，因为那样就成传统的相似度计算方法了，由此我们也可以推知λ不能太小，小到接近 0，那样就没有强调项目类型的效果了，也就丧失了我们提出的计算项目相似度方法的意义了。

综上所述，λ参数应该是在(0，0.5)的一个值，这个值不能过小。下文我们通过实验来验证我们这里讨论是否正确，并通过调整该参数的值来观察其对算法的推荐质量效果的影响。

3.3 评价标准

推荐系统最感兴趣的是预测或者推荐的质量。协同过滤算法的主要目的是预测用户未评分项目的评分值，预测值的精确度是评价衡量推荐系统质量的主要度之一，推荐系统推荐质量的评价标准主要包括统计精确度度量方法和决策支持精确度度量方法两类[7,11]。

1) 统计精确度度量

通过系统对目标项目的预测值与用户对目标项目的实际评价值进行比较，来评价系统预测的准确性。平均误差(Mean Absolute Error,MAE)是使用较早的一个准确性评价标准。其它的标准还有 root mean squared error(RMSE)，correlation between ratings and predictions 等。平均绝对偏差 MAE 是用于度量预测值与实际评价值之间的偏差的方法，MAE 越小，预测精度越高，推荐质量越高。设预测的用户评分集合表示为$\{p_1,p_2,...p_N\}$，对应的实际用户评分集合为$\{q_1,q_2,...q_N\}$则平均绝对偏差 MAE 定义为：

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{3.7}$$

2) 决策支持精确度度量

用于评价预测对帮助用户从众多项中选择到满足他们兴趣的项的有效性。这个标准基于这样的假设预测过程即用户的决策的过程，即用户认为该项好或不好，用户选择该项或不选择该项。根据以上假设，对于按 1-5 标准对项进行评价，如果用户仅选择预测值为 4 或者预测值更高的项目，则预测值为 1.5 和 2.5 已没有什么区别。常用的标准有 reversal rate、weighted errors 以及 ROC sensitivity。

由于统计精度度量方法中的平均绝对偏差 MAE 易于理解，可以直观地对推荐质量进行度量，是最常用的一种推荐质量度量方法，本文以下所有实验主要采用平均绝对偏差 MAE 作为度量标准。

3.4 数据集

实验使用的数据来自 Minnesota 大学进行 GroupLens Research 项时收集的 MovieLens 数据集。MovieLens 是基于 Web 的研究性的推荐系统，注册用户必须至少对它所拥有的影片中的 15 部进行评价才可以使用该系统。从该站点下载 1997 年 9 月 19 日到 1998 年 4 月 22 日的数据集，该数据集已经完成数据清理，其中投票数小于 20 的用户已经被清除。数据集为 943 个用户对 1682 个项 (影片)的 10 万条投票记录，用户评分数据集的稀疏等级为 1-100000/(943×1682) = 0.9370。其中有：(1)u.data，为未排序的全部 10 万条投票记录，结构为(用户 ID，项 ID，投票分数，时间戳)；(2)u.user，为用户信息；(3)u.item，为项(影片)信息，包括影片的名称、发行年份和分类等信息；(4)u.genre，为项分类信息，列出了 19 个分类的具体名称，包括动作、冒险、动画、儿童、喜剧、科幻、恐怖、战争等；(5)u1.base 到 u5.base，以及 u1.test 到 u5.test 五组文件，分别为把 u.data 中的记录按照 80%和 20%的比例，进行记录分割得到的训练集和测试集。

本文所有实验均采用整个数据集的 80%作为训练集，20%作为测试集。

3.5 实验与分析

这一节将呈现项目类型优化的基于项目协同过滤算法的优化结果。我们将进行一系列相关实验，并对实验结果进行分析。

3.5.1 实验方案

我们的实验主要分两部分，以平均绝对偏差(MAE)为主要度量标准：

(1) 以传统相似性度量方法分别对同一数据集同一训练集和测试集比例进行实验，

选择结果最佳的方法。

(2) λ参数对本章算法的效果影响，通过调整λ的值，验证传统相似度和类型相似度的对推荐质量的影响的大小，并确定λ参数的最佳值；在不同邻居个数情况下，将本章算法与传统基于项目的协同过滤算法进行对比的实验。

3.5.2 实验结果

3.5.2.1 传统相似度度量方法的选择

正如本章第一节所介绍的传统基于项目的协同过滤算法的相似性度量主要有三种方法，余弦相似性、修正余弦相似性和相关相似性，我们经过试验，计算其 MAE，试验结果如图 3.2。

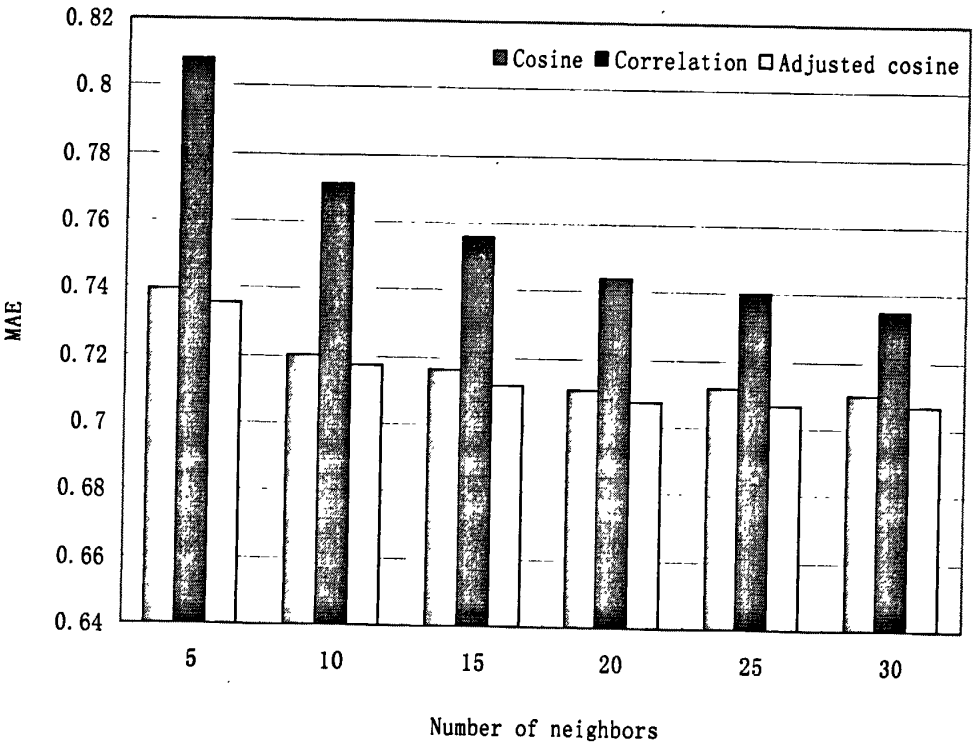

图 3.2 相似性度量标准比较

Fig. 3.2 Comparison of similarity measure methods

由图 3.2 可知，在不同邻居个数的各种情况下，修正余弦相似性度量方法的 MAE 低于其他两种方法，因此我们选择修正余弦相似性作为计算项目间的传统相似性的方法，本章以下实验均如此。

3.5.2.2 参数λ的影响

下面我们通过变化λ参数的值来观察该参数对我们提出的算法的预测效果的影响，并由此知道参数λ的最佳值，为了更好的观察其影响，这里我们分别取最近邻居个数为

10、20、30，结果如图 3.3。

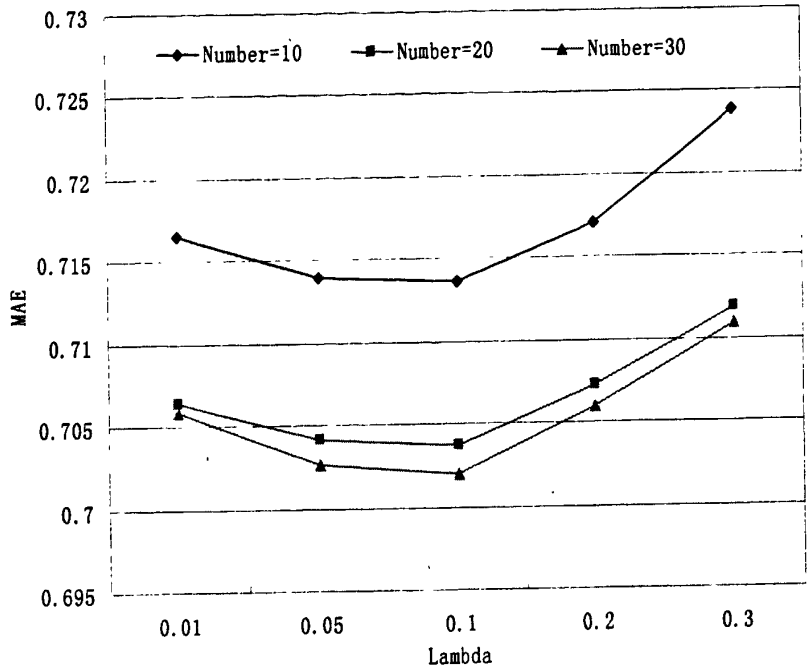

图 3.3 参数 λ 的影响

Fig.3.3 The impact of parameter λ

由图 3.3 我们可知，在不同的邻居个数下，随着参数 λ 的值的增大,MAE 的值的变化趋势是相同的，当参数 λ 小于 0.1 时，参数 λ 越大，MAE 的值越小，而当参数 λ 越过 0.1 后，其值越大，MAE 的值却越大。这样随着参数 λ 的值增大，推荐效果是先变好后又变坏，这与我们之前在 3.2.4 节讨论的 λ 的范围吻合，即 λ 参数是在（0，0.5）的一个值，这个值不能过小。从图上我们看到在不同的项目邻居个数下，参数 λ 的值为 0.1 时推荐效果均为最好。

3.5.2.3 与传统基于项目的协同过滤相比

为了检验本章提出的基于项目协同过滤类型优化算法的有效性，我们以传统的未优化的基于项目协同过滤推荐算法作为对照，这里传统的基于项目协同过滤算法的相似性度量方法以及产生推荐方法均选前文实验结果中佳者，而我们提出的算法涉及到的部分也一样。根据上一节实验的结果，这里我们取参数 λ=0.1，实验结果如图 3.4 所示。

由图 3.4 可知，本章的基于项目协同过滤的类型优化算法在不同项目邻居个数情况下，都具有较小的 MAE。由此可见，与传统基于项目协同过滤推荐算法相比，我们提出的基于项目协同过滤的类型优化算法明显提高了预测精度，可以提高推荐系统的推荐质量。

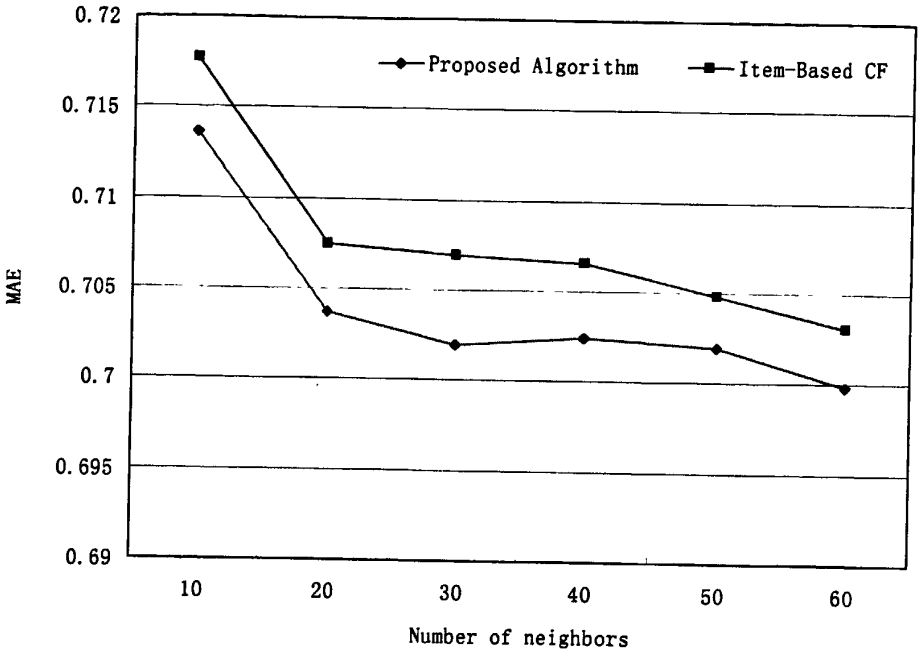

图 3.4 算法比较

Fig. 3.4 Comparison of algorithms

3.5.3 实验结果分析

由图 3.4 我们可以看到，与传统基于项目协同过滤算法相比，我们提出的基于项目协同过滤的类型优化算法的确获得了较好的推荐效果，这充分说明项目类型在对项目间相似性度量中需要加强，这提高了项目间的相似度度量的精确度。同时从图 3.3 中，我们注意到参数 λ 的值较小但不是太小时，我们的算法才能获得较好的预测效果，这也验证了我们之前关于参数 λ 取值范围的讨论，说明项目的类型相似性只需加强，但不能抹杀其他因素对项目相似性的作用。

3.6 小结

本章通过分析传统基于项目协同过滤推荐算法中的一些不足，说明项目类型信息对项目间相似性度量的重要性，提出在传统基于项目协同过滤推荐算法只利用用户评分数据的基础上引入项目类型信息，利用该信息参与计算项目间的相似度的想法。通过引入项目类型矩阵计算项目间的类型相似度，并与传统基于用户评分矩阵相似度计算结果通过线性方式相结合，共同作为项目间的相似性，以期减少评分矩阵稀疏对相似性度量的影响，提高相似度度量的精确性，从而改善推荐算法的推荐质量。最后通过具体实验确定算法中主要步骤的具体实现方法。实验结果表明，我们得到在基于项目协同过滤算法中使用项目类型优化取得了很好的推荐质量，比传统的基于项目的协同过滤推荐算法效果好，提高了预测的精确度，改善了推荐质量。

第四章 基于用户协同过滤的类型优化算法

4.1 传统基于用户协同过滤算法

基于用户的协同过滤是个性化推荐中应用最为广泛的方法之一，与基于项目协同过滤相比，基于用户的协同过滤技术更早地就被提出来了，基于用户协同过滤基于这样一个假设：如果用户对一些项目的评分相似，则他们对其它项目的评分也会比较相似。算法的基本思想是：目标用户对未评分项目的评分，可以通过其最近邻居对该项目的评分来逼近。

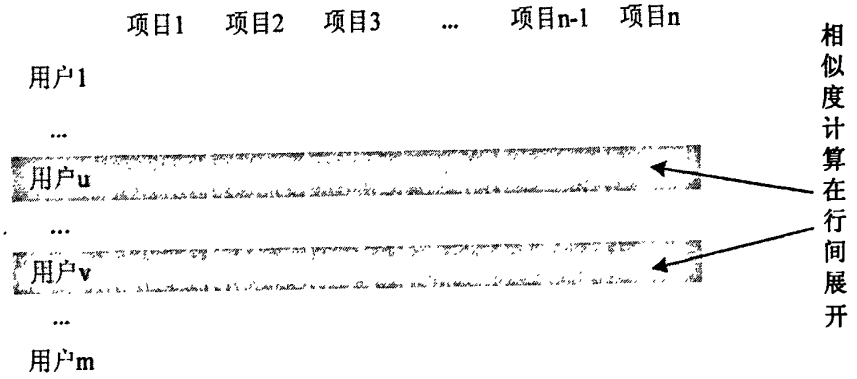

图 4.1 基于用户协同过滤相似度计算

Fig.4.1 Similarity computing of user-based CF

基于用户的协同过滤推荐的优势是很明显的。首先，它能够通过用户间的相互协助、根据用户对项目的评价的相似性对用户进行分类，找到目标用户的邻居。这样能得到的推荐结果是比较精确的。其次，在基于用户的系统过滤系统中，所有用户都能从邻居用户的评价中受益，只要每个用户为系统贡献一份力量，系统就能维持比较好的性能，这就是角色一致性(role uniformity)[32]。角色一致性能推动协同过滤系统良性发展，使系统保持有效的推荐。最后，基于用户的协同过滤系统容易挖掘出目标用户潜在的新兴趣，即能够实现奇异发现。

虽然基于用户的协同过滤技术作为一种典型的推荐技术有其相当广泛的应用，但它仍有许多的问题需要解决。最典型的问题就是数据的稀疏性问题，在电子商务系统中，一般用户很多且项目也很多，但通常情况下，每个用户只会对很少的项目做出评价，所以在用户－项目矩阵中，每行只有很少的项目有评价数据，整个数据阵异常稀疏。根据对一些大型电子商务系统的调查结果，系统中的用户的评价数据一般都在 1%以下。在这种情况下得到的用户间的相似性是非常不准确的，那么寻找到的邻居也就不太可靠，不可靠的邻居的推荐的将严重影响系统的推荐效果。另外，在当前的使用基于用户的协

同过滤的推荐系统中，由于其算法一味地追求推荐的准确性，常常造成用户得到的推荐种类单一项目雷同，无法真正满足用户多样的购买需求，局限了用户在商品购买上的视野。此外，算法的运算性能在基于用户的协同过滤推荐也是必须考虑的方面。当用户规模和项目规模非常庞大的时候，计算用户间的相似性寻找邻居将是一件非常费时的事情，如果计算时间过长，无疑将影响在线推荐的实时性。

基于用户的协同过滤算法实现步骤和基于项目协同过滤基本一致：计算用户之间的相似程度，确定最近邻居集和产生推荐。其中计算用户之间的相似性与基于项目协同过滤不同的是，其相似性度量是在用户评分矩阵的行间进行的，如图 4.1。下面我们将进行进一步说明。

4.1.1 相似度计算

如何度量用户与用户之间的相似程度？同样也主要有余弦（cosine）相似性、修正余弦(adjusted cosine)相似性和相关(correlation)相似性。

(1) 余弦相似性：用户评分被看作为 n 维项目空间上的向量，这与项目相似性计算中的余弦相似性计算公式一样，通过向量间的余弦夹角度量，只是将向量变为用户在项目空间的评分向量。设用户 i 和用户 j 在 n 维项目空间上的评分分别表示为向量 $\vec{i}$、$\vec{j}$，代入公式 3.1 即可。

(2) 修正余弦相似性：由于在余弦相似性度量方法中没有考虑不同用户的评分尺度问题，例如有些人常给高分，而有些人给的评分普遍较低。为了克服这一缺陷，修正余弦相似性方法中减去用户对项目的平均评分。设经用户 i 和用户 j 共同评分的项目集合用 I_{ij} 表示，I_i 和 I_j 分别表示经用户 i 和用户 j 评分的项目集合，则用户 i 和用户 j 之间的相似性 $sim(i,j)$ 为：

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-\overline{R}_i)(R_{j,c}-\overline{R}_j)}{\sqrt{\sum_{c\in I_i}(R_{i,c}-\overline{R}_i)^2}\sqrt{\sum_{c\in I_j}(R_{j,c}-\overline{R}_j)^2}} \tag{4.1}$$

(3) 相关相似性：设经用户 i 和用户 j 共同评分的项目集合用 I_{ij} 表示，则用户 i 和用户 j 之间的相似性 $sim(i,j)$ 通过 Pearson 相关系数度量为：

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-\overline{R}_i)(R_{j,c}-\overline{R}_j)}{\sqrt{\sum_{c\in I_{ij}}(R_{i,c}-\overline{R}_i)^2}\sqrt{\sum_{c\in I_{ij}}(R_{j,c}-\overline{R}_j)^2}} \tag{4.2}$$

4.1.2 最近邻居

根据计算好的用户之间的相似性，对于目标用户 u，按照其与其他用户的之间的相似性从大到小排序，产生一个最近邻居候选集合 $C=\{N_1, N_2, ..., N_t\}$，u 不属于 C。再根据

要预测该用户对项目 i 的评分，那么从候选集合中选择对项目 i 有评分的用户组成相应的最近邻居集合。

4.1.3 产生推荐

根据最近邻居集合 N 产生推荐，预测用户 u 对项目 i 的评分 $P_{u,i}$，对于基于用户协同过滤，以下公式较为常用并取得较好效果：

$$P_{u,i} = \overline{R}_u + \frac{\sum_{v \in N} sim(u,v) \times (R_{v,i} - \overline{R}_v)}{\sum_{v \in N} |sim(u,v)|} \tag{4.3}$$

其中 $R_{v,i}$ 表示用户 v 对项目 i 的评分，$\overline{R}_u$、$\overline{R}_v$ 分别表示用户 u 和用户 v 对以评分项目的平均评分，$sim(u,v)$ 是用户 u 和用户 v 的相似度。

若要推荐项目，可通过上述方法预测用户对所有未评分项目的评分，然后选择预测评分最高的前若干个项目作为推荐结果反馈给当前用户。

4.2 基于用户协同过滤的类型优化算法

4.2.1 问题提出

由上一章的分析和实验我们看出项目类型对传统基于项目协同过滤算法的优化显著效果显著，那么其在基于用户协同过滤算法中是否能改善效果呢？

在 3.2.1 节中我们已经提到，大多数推荐系统都会对自己提供的项目进行分类，其实这样做是为了方便用户更好地找到自己可能喜欢的影片，这也正好说明影片类型对用户的选择的影响力。我们都有这样的经验，有些人喜欢看动作片，讨厌看恐怖片，如果一部电影属于动作片，他往往会有兴趣，而如果一部影片是恐怖片，那他去看的可能性会较低，或者会不太喜欢。可见项目分类信息对用户兴趣的指向性。所以项目分类信息是对于个性化推荐的质量的影响是较为重大的。

我们再以看电影为例，用户 A 看了一部悲剧影片，A 给了较高的评分，那么你是不是会想当然的认为他喜欢悲剧电影呢？但 A 通常还是喜欢看含有喜剧成分的电影，那他为什么会看呢？也许是因为有他喜欢的演员或者导演或者某位爱看悲剧的朋友劝说他去看的，而他因为这部电影的确拍摄的不错或者演员的演技不错，所以给了高分，这里不确定因素很高，所以单看评分是不够全面的。而且 A 看的含喜剧的电影比较多，但他并不见得对每部含喜剧的电影都打分较高，以至于他看其他不含喜剧的电影打分可能比一些喜剧电影还要高，在用户对评分的项目稀少的情况下，这很有可能会导致我们使用传统方法计算相似用户时出现误差，而且你又如何得知 A 喜欢喜剧电影？那也是通过电影的类型信息联系起来的。所以我们可以通过利用项目类型探索用户更深层次的兴趣所在，需要强调项目类型信息在相似性计算中的比重，因为它比起其他那些因素能更稳定、

更好地体现用户的喜好。

假设还有另两个用户 B、C，我们依照传统的通过用户评分矩阵相似性度量方法计算用户 A 与用户 B 的相似度 $sim(A,B)$ 以及 A 和 C 的相似度 $sim(A,C)$，而 $sim(A,B)=sim(A,C)$，如果 B 和 C 评分的所有项目相同，评分也相同，那么这自然没有必要使用项目类型信息了，但实际中这种情况几乎不存在。那么就是 B 和 C 用户并不是评分的项目完全相同而导致这种与 A 相似性相等的结果，例如 B 与 A 有一些相似的项目，这些项目含有喜剧因素的较多，C 与 A 有一些相似的项目，项目含有动作因素的较多，那么我们之前提过 A 还是更喜欢看喜剧电影，那么谁与 A 更相似呢？自然是 B。而传统方法计算的结果显然不是实际应有的结果。

综上所述，我们也可以发现传统的基于用户协同过滤推荐算法也存在与上一章传统基于项目协同过滤算法类似的问题。在计算用户相似度时，它也只考虑用户评分矩阵，没有利用项目类型信息，这样计算的用户间相似性是不准确的，也使得推荐时缺乏个性化。所以我们需要在用户评分的基础上，加强针对用户对于项目类型的较为深层的相似性的计算，这可能会带来一些收获。

4.2.2 用户-类型矩阵

如何在基于用户的协同过滤算法中使用项目类型信息呢？这里不如上一章基于项目协同过滤算法中那样直接简单，如何建立用户和项目类型间的关联？这里通过用户评分矩阵和项目类型矩阵转换得出用户-类型矩阵，具体生成过程如下：

算法 4.1 用户-类型矩阵生成

输入：用户评分矩阵 R，
项目类型矩阵 G；
输出：用户项目矩阵 UG。

```
for :ug_{u,k} ∈UG                    /*初始化 UG*/
        ug_{u,k} = 0;
for :r_{u,i} ∈R {
    if(r_{u,i} != 0){                /*用户 u 对项目 i 有评分*/
     for :g_{i,k} ∈G_i               /*G 的第 i 行，即项目 i*/
        ug_{u,k} += g_{i,k};
    }
}
```

得到的用户类型矩阵如表 4.1。

表 4.1 用户类型矩阵

Table 4.1 The matrix of user and genre

	类型 1	类型 2	类型 3	类型 4	…	类型 k
用户 1	0	1	2	4	…	0
用户 2	2	0	1	0	…	3
用户 3	0	4	3	1	…	0
…	…	…	…	…	…	…
用户 m	0	2	0	0	…	0

用户矩阵一定程度上较为直观地体现了用户的偏好，以项目为电影为例，从用户对各类型影片的偏好，我们可以得出用户比较喜欢含有什么类型的影片，例如通过用户类型矩阵得出，用户对应动作类型的值在所有类型中最高，说明该用户喜欢动作片，那么如果一部电影是动作片，他喜欢的可能性就较大，当然他可能因为主演他不喜欢而不去看，但从普遍意义上，该用户还是喜欢动作片的，而这也较其他因素更重要更稳定的体现了用户的喜好。

4.2.3 类型优化

利用上一小节产生的用户类型矩阵计算用户相似性来修正传统相似性度量方法，针对用户-类型数据的特性，其相似性计算我们依然采用余弦相似性度量，因为用户类型数据并没有什么针对不同用户有所不同之说。

$$sim(u,v)_{user-genre} = \cos(\vec{u},\vec{v}) = \frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|\cdot\|\vec{v}\|} \tag{4.4}$$

这里 $\vec{u}$、$\vec{v}$ 分别为用户 u，v 在用户-类型矩阵对应的行向量，分别代表了用户 u 和用户 v 评分的项目所含的类型。

从前文的分析，对于计算用户相似性，我们也只许强调项目类型因素。因此，我们保留传统基于用户协同过滤推荐算法仅基于用户评分数据的相似性度量，再加入用户-类型矩阵计算的相似度。这里我们依然采用线性结合方式，最终项目 u 和 v 的相似性计算公式为：

$$sim(u,v) = (1-\lambda)\times sim_R(u,v) + \lambda \times sim_{user-genre}(u,v) \tag{4.5}$$

其中 λ 是一个在[0，1]范围内的参数，该参数允许我们设定用户间相似性依赖于传统相似性和类型相似性的比重。同时也一定程度上平衡用户评分矩阵与用户-类型矩阵的疏密性的差异，注意：$sim_R(u,v)$ 和 $sim_{usergenre}(u,v)$ 虽然都是计算用户 u 和 v 之间的相似性，但是基于不同的数据计算的，前者是使用用户项目评分矩阵，而后者是使用用户

类型矩阵。

这里我们也讨论一下参数λ的范围，即类型相似性与传统用户相似性之间所占的比重，由于用户评分数据无疑已经包含很多与用户相关联的信息，而推荐系统是以人为服务对象，评分矩阵不止包含了用户的喜好和项目之间的关联，而类型信息这个项目本身信息也隐式地包含在其中，我们所要的是要加强类型对相似性度量的贡献，因为其对于分析用户的喜好是很重要的，所以只需加重类型的比重，而不能完全抹杀传统相似性度量中其他因素的作用（如用户的个人特征，性别、国籍等）。与在上一章的讨论类似的，传统基于用户评分矩阵计算的相似度占比重应该较大，而且λ也不能为 0，也不能接近 0，那样用户-类型矩阵计算出的相似度的部分与λ相乘后就对用户相似性的度量没有了效力，我们方法的优化性也就显现不出来了，由此可以推知，λ在(0,0.5)范围中,应是一个较小的值，这个值不能过小。下文我们通过实验来验证我们这里讨论是否正确，并通过调整该参数的值来观察其对算法的推荐质量效果的影响。

4.3 实验与分析

这一节将呈现项目类型优化在基于用户协同过滤算法的优化结果。我们将进行一系列相关实验，并对实验结果进行分析。

4.3.1 实验方案

我们的实验主要分两部分，数据集以平均绝对偏差(MAE)为主要度量标准：

(1) 以传统相似性度量方法分别对同一数据集和同样的训练集测试集比例进行实验，选择结果最佳者；

(2) λ参数对本章算法的效果影响，通过调整λ的值，验证传统相似度和类型相似度的对推荐质量的影响的大小，并确定λ参数的最佳值；在不同邻居个数情况下，将本章算法与传统基于项目的协同过滤算法进行对比的实验。

4.3.2 实验结果

4.3.2.1 传统相似度度量方法的选择

将本章第一节中提到的三种用户间相似度度量方法应用于同一数据上，进行比较试验，结果如图 4.2。

由图 4.2 可知，修正余弦相似性度量方法的 MAE 低于其他两种方法，因此我们选择修正余弦相似性作为计算用户间的传统相似性的方法，本章以下实验均如此。

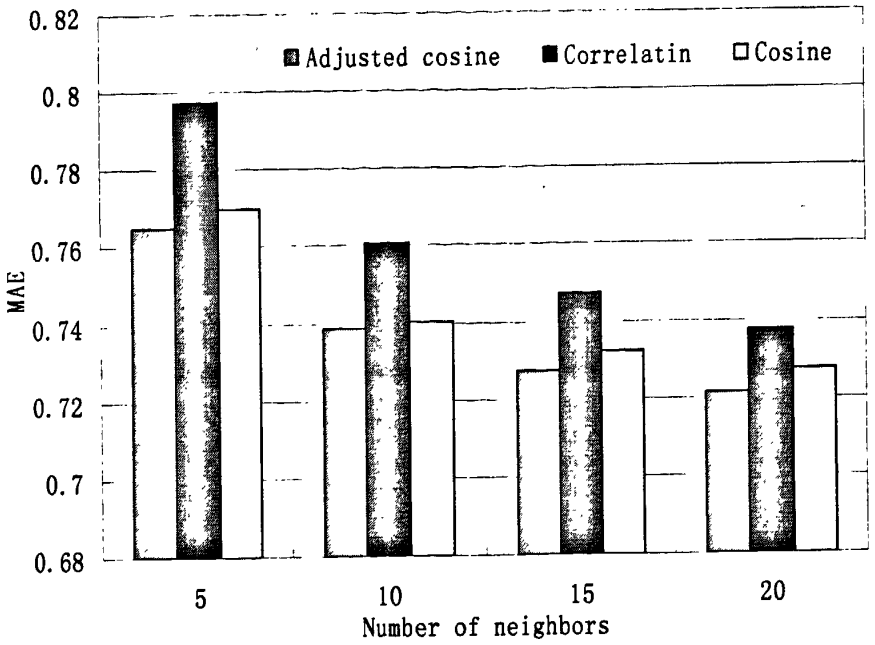

图 4.2 相似性度量标准比较

Fig. 4.2 Comparison of similarity measure methods

4.3.2.2 参数 λ 的影响

下面我们通过变化 λ 参数的值,来观察该参数对预测的准确性的影响，并由此知道参数 λ 的最佳值，为了更好的观察其影响，这里我们分别取最近邻居个数为 15、20、25，结果如图 4.3。

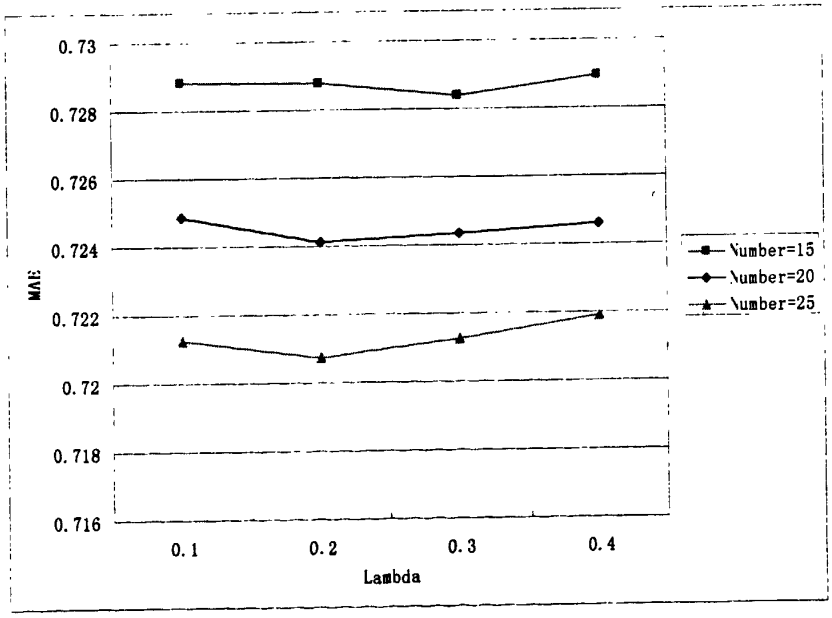

图 4.3 参数 λ 的影响

Fig. 4.3 The impact of parameter λ

由图 4.3 我们看到当目标用户邻居数为 15 时，参数 λ 为 0.3，MAE 的值最小；而当

目标用户的邻居个数为 20、25 时，参数 λ 为 0.2，MAE 的值最小；有此可知，参数 λ 的最佳值在 0.2 和 0.3 之间。

4.3.2.3 与传统基于用户的协同过滤相比

为了检验本文提出的算法的有效性，我们以传统的未优化的基于用户协同过滤推荐算法作为对照，这里传统的协同过滤的相似性度量方法以及产生推荐方法均选前文实验结果中佳者，而我们提出的算法涉及到的部分也一样。根据上一步实验结果，这里取参数 λ=0.25，实验结果如图 4.4 所示。

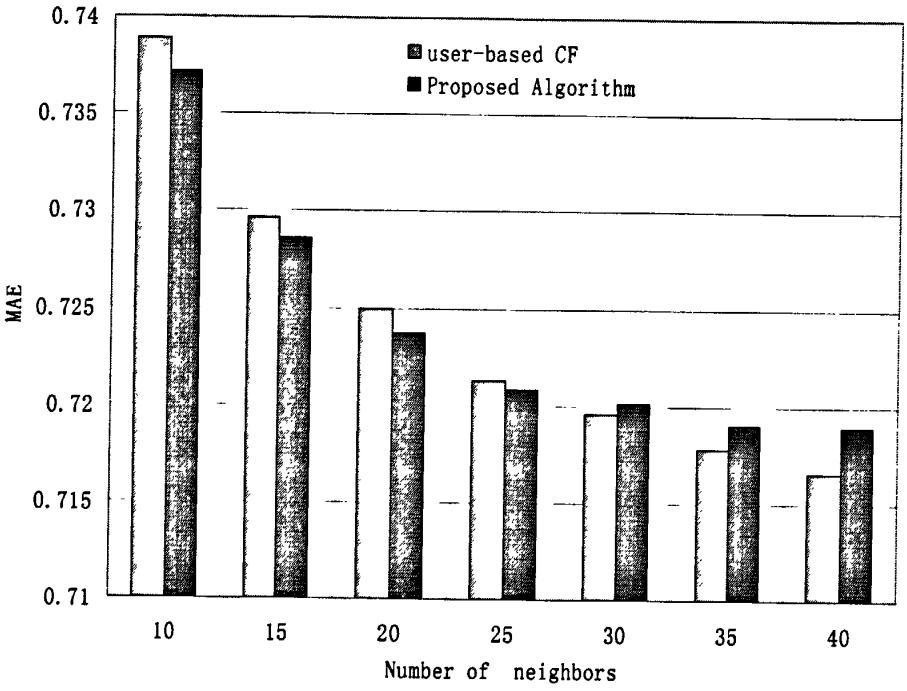

图 4.4 算法比较

Fig. 4.4 Comparison of algorithms

由图 4.4，我们可以看到，在邻居用户个数为 30 以前，我们提出的类型优化算法的 MAE 的值比传统的基于用户协同过滤算法小，说明推荐质量较好；而当用户邻居个数为 30 或更大后，我们提出的算法的 MAE 反而比未优化的基于用户协同过滤算法大了，而且这种增大随着用户个数的增大而越发明显。

4.3.3 实验结果分析

由实验结果，我们看到，但邻居用户个数增加到一定数目时，基于用户协同过滤的类型优化算法的推荐效果变得差起来，比基于用户协同过滤算法还要差，这是为什么呢，用户评分矩阵和用户类型矩阵的行都是用户，个数一样，但我们注意到用户—类型矩阵，其列为项目类型，个数是基本固定不变而且很小的，本实验中根据数据集中的 u.genre

文件（参见3.4节）为19种；而用户评分矩阵的列为项目，项目的数量很大，在本章实验中为1682个项，可见两者的稀疏度的巨大差异，这一点虽由我们通过使用参数λ平衡掉了一部分，但是由于使用数据集的用户评分数据的稀疏性加大，如图4.5，该图为我们所使用的训练集数据的评分分布情况，我们看到低于80个评分的用户就占总体用户近2/3的比例，而高于80个评分的用户仅占总体用户个数的1/3多一点，当用户的邻居个数增加到一定程度，就会从占总体用户近2/3的评分个数极少的用户中选择，而其与目标用户的相似性由于评分个数少必然比之前的用户小很多。这样随着目标用户的邻居个数的增加，邻居用户的评分项个数将越来越少，而用户—类型矩阵的疏密却没有太大的变化，这是由于项目类型本来就数目较小，而每个用户都会有一些偏好的项目类型，其多少差别不大；这样利用用户类型矩阵计算的相似度不会随最邻居个数的增加有何明显变化的趋势，而通过用户评分矩阵计算的却因矩阵变得稀疏而导致相似度急剧降少，从而打破了参数λ建立的平衡，平衡传统相似度和类型相似度比重和平衡用户评分矩阵与用户-类型矩阵的比例，但这个问题可以通过进一步减小参数λ的值来得以解决。但我们实验发现减小参数λ的值，能减少我们的算法与传统基于用户的协同过滤间的MAE差距，当我们将参数降低到极小时，用户类型相似性在用户相似性计算中占的比例又太小，而没有明显的改进了。

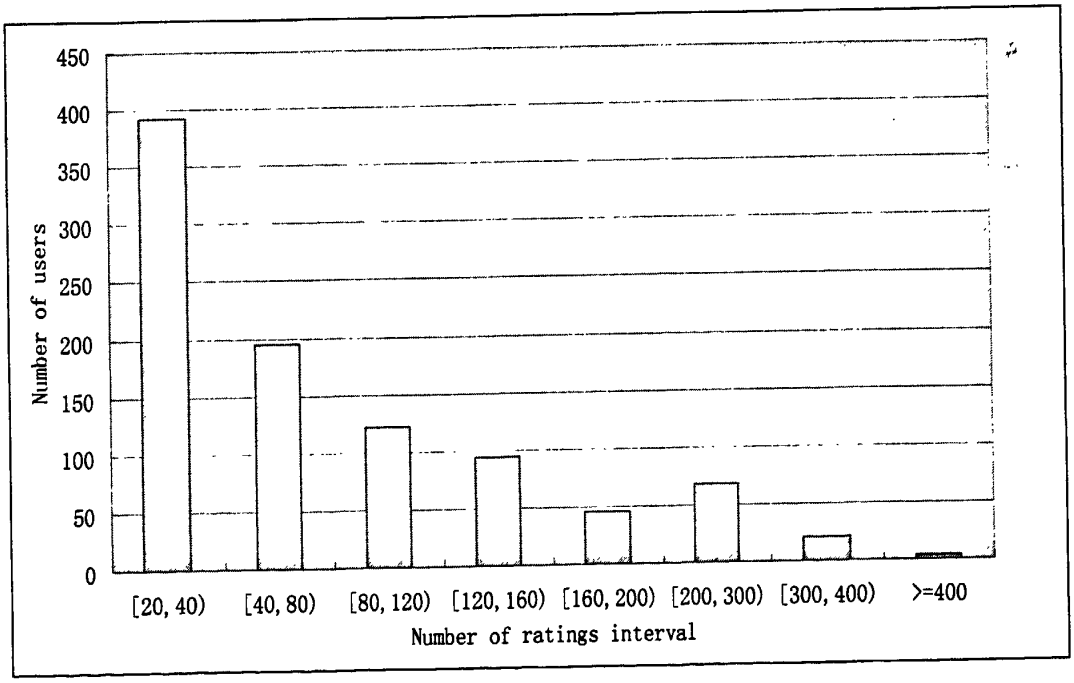

图 4.5 用户评分分布

Fig. 4.5 Distribution of users' rating

我们也同注意到图5.4中即使是之前当用户个数没有超过25时，本章的算法也没有得到如上一章基于项目协同过滤的类型优化算法的那样明显提高推荐质量的效果，就其

原因是由于用户类型毕竟是由用户评分和项目类型矩阵结合而生成的，没有像项目类型数据直接作用于基于项目协同过滤的项目相似度计算，而是间接的，这必然使得效果不是很明显，而且其生成方法简单。可以考虑通过用户对的项目评分对其中含有的类型赋予不同的权重的方法改善这个问题。

4.4 进一步的构想

通过上一章和本章的实验，我们看到基于项目类型优化的协同过滤算法一定程度上提高了传统基于用户和基于项目的协同过滤算法的推荐质量，那么在其他协同过滤算法中是否也能利用项目类型信息提高推荐质量呢，这一点有待进一步研究。在看到实验结果较好时，我们也注意到类型优化在基于项目的协同过滤中的使用效果比在基于用户协同过滤算法中更明显，这是由于项目类型是项目的本身属性，使用项目类型直接作用于基于项目的协同过滤算法的项目相似度计算，比通过项目类型和用户评分矩阵经过转换的用户类型矩阵这样间接地作用于基于用户协同过滤算法的用户相似度计算影响更大，由此我们想到，如果我们也可以从用户信息中找到这样的信息加以利用，使用该信息直接作用于基于用户的协同过滤算法的相似度计算，那么我们可以预期能得到更为显著的影响，我们大胆地期待这也能一定程度上提高推荐质量，同时我们亦可以将这一信息通过用户评分数据进行转换，在作用于基于项目协同过滤算法，提高推荐质量，因为什么样的项目会被什么样的用户所喜欢，这个信息对推荐无疑是有着重大意义的，通过利用用户信息，可以更直观地观察喜欢某项目的用户的共同特征，加强其在计算相似度中的比例，自然会提高计算项目相似度的精确度，从而改善推荐质量。

4.5 小结

本章通过分析传统基于用户协同过滤推荐算法中的一些不足，提出在传统基于用户协同过滤推荐算法只利用用户评分数据的基础上引入项目类型信息，通过用户评分矩阵和项目类型矩阵转化得到用户-类型矩阵，通过用户-类型矩阵计算用户间的类型相似度，并与传统基于用户评分矩阵相似度计算结果通过线性方式相结合，共同作为用户间的相似性，以期减少评分矩阵稀疏对相似性度量的影响，提高相似度度量的精确性，从而改善推荐算法的推荐质量。最后通过具体实验确定算法中主要步骤的具体实现方法。实验结果表明，基于用户协同过滤类型优化算法一定程度上提高了推荐质量，即在用户邻居个数不超过一定数目时，比传统的基于用户的协同过滤推荐算法效果好，提高了预测的精确度。然而随着用户邻居个数增多，其效果反而比传统基于用户协同过滤算法差。通过分析，我们得出是由于用户评分个数急剧减少，打破了参数λ的构造的平衡，减小参数λ到过小，就没有明显改善效果。而且用户-项目类型矩阵对于用户来说的间接性也是

类型优化方法的效果没有基于项目协同过滤中那么明显。通过实验结果，我们还是可以看出，在传统基于用户协同过滤推荐算法中引入项目类型信息的有一定必要性和重要性。

第五章 改进的基于用户协同过滤算法

5.1 组合推荐

5.1.1 组合推荐技术

每种推荐技术都有各自的优缺点。为了给用户提供更准确更合理的推荐，我们可以在设计推荐系统的推荐方法模块时结合多种基本推荐技术，以达到扬长避短的目的。这就是组合推荐的思路。

目前并没有一种非常完美的推荐方法。所以要实现一个现实的推荐系统，组合推荐的思路非常必要。一般而言，推荐技术的组合有以下几种思路[39]。

(1) 加权(Weight)：采用多种推荐技术得到对某一项目的预测评分，根据权重相加得到总评分，以此得出推荐结果。

(2) 切换(Switch)：具体采用哪种推荐技术取决于当时的实际情况，根据应用场合切换不同的推荐技术。

(3) 混合(Mixed)：同时采用多种推荐技术进行推荐。

(4) 特征组合(Feature combination)：组合来自不同推荐数据源的特征并被一种推荐算法所采用。

(5) 层叠(Cascade)：一个推荐器从另一种推荐器中提炼抽取一部分推荐。

(6) 特征放大(Feature augmentation)：一种推荐技术的输出结果作为另一种推荐技术的特征输入。

(7) 模型放大(Meta-level)：被一种推荐器学习的模型作为另一种推荐器的输入。

目前研究比较多的推荐组合有基于内容推荐和协同特征增量的组合推荐，基于协同和基于内容的组合推荐，基于协同和人口统计的组合推荐等。这里我们考虑如何将基于用户协同过滤和基于项目协同过滤进行组合，有些研究工作者已在此方面做一些工作，下一节将作简要介绍。

5.1.2 基于项目和基于用户协同过滤算法组合推荐

正如我们在第二章中提到，协同过滤根据所参考的事物相关性来分，有基于用户和基于项目两种。

基于用户协同过滤是最先提出来的，应用比较普遍，首先，它能够通过用户间的相互协助、根据用户对项目的评价的相似性对用户进行分类，找到目标用户的邻居。这样能得到的推荐结果是比较精确的。其次，在基于用户的系统过滤系统中，所有用户都能从邻居用户的评价中受益，只要每个用户为系统贡献一份力量，系统就能维持比较好的

性能，这就是角色一致性(role uniformity)。角色一致性能推动协同过滤系统良性发展，使系统保持有效的推荐。最后，基于用户的协同过滤系统容易挖掘出目标用户潜在的新兴趣，即能够实现奇异发现。但这种方法在实践过程中遇到一个主要问题就是稀疏性，也就是指在系统运行过程中，由于项目数量较大，用户评分数量少而造成用户—评分矩阵的稀疏，导致用户间的相似性计算不准确，得到的邻居用户也就不可靠，所以基于用户的协同过滤方法很难利用这些评价来发现相似的用户。

对于基于用户的协同过滤的缺陷，基于项目的协同过滤有着行之有效的解决方法。事实上，基于项目的协同过滤最早提出来的时候，就是为了解决传统的基于用户的协同过滤的稀疏问题。 基于用户的协同过滤系统运行的瓶颈是要在一个很大的用户群中找出合适的邻居，基于项目的协同过滤系统可以通过寻求项目之间的相似关系，而不是用户之间的相似关系来避免这个瓶颈问题。因为在典型的电子商务环境中，项目之间的关系相对来说比较稳定，因此可以离线完成工作量最大的相似性计算步骤，所以利用项目之间的相似性，基于项目的协同过滤算法可以花费较少的在线计算时间来得到与基于用户的协同过滤系统准确性相近甚至更好的预测结果[11]，这种方法在某种程度上解决了基于用户的协同过滤系统中存在的可扩展性问题。但是基于项目的协同推荐不能作出“跨类型”的推荐，因为它推荐的总是相似的项目，也就不能挖掘用户的潜在兴趣。

鉴于两种方法各有其优点和缺点，近些年，将基于用户和基于项目协同过滤二者结合起来的研究也开始多起来，文献[40]就是利用基于项目协同过滤算法填充用户评分矩阵，降低矩阵的稀疏性，弥补基于用户协同过滤由于评分矩阵稀疏导致推荐质量不高的问题，然后在以填充的评分矩阵上使用基于用户的协同过滤进行推荐，取得了不错的效果。但其没有考虑项目的类别，从而影响了推荐质量，文献[41]考虑了类别性，在利用基于项目协同过滤前使用聚类算法先项目进行聚类，再在簇内使用基于项目协同过滤算法进行填充。还有文献[42]中使用线性结合的方式，将基于用户和基于项目的预测结果结合起来，参数通过测试得出最佳值，但目标用户的已知评分项的个数是多样化（而该文章是规定了目标用户的已知评分项的个数，分别进行实验）以及使用的相似度量和产生推荐的方法的不同，参数值都可能发生巨大的变化，那么我们就无法一一事先测定的参数的最佳值或较好的值，从而使其使用的有一定的局限性，所以对于实际推荐系统来说，这是难以实现的。

本文就考虑利用组合推荐的思想，将基于用户和基于项目协同过滤结合起来解决协同过滤中存在的问题。

5.2 改进的基于用户协同过滤算法

5.2.1 算法的提出

传统的协同过滤推荐算法，往往把用户对所有项目的评分整体作为一个向量，进行相似度的计算和指导最近邻居的选择，存在如下缺点：

采用这种方法得到的用户相似性，反映的是用户对所有类型项目的偏好相似性。然而，现实中用户存在不同的爱好和倾向，即对不同类型项目的关注有所不同，几乎不可能出现在所有类型项目上具备共同偏好的相似用户。因而这种方法得到的相似性结果不具备代表性。

由于相似性没有体现项目类别性，使得推荐时缺少个性，难以适应目前电子商务系统日趋多样性和个性化的趋势。

这种情况实际中常有，为了便于说明问题，我们假设与用户 A 相似的用户有 B、C,他们与 A 的相似度大小依次的顺序为 B、C,其中 A 喜欢动作片、喜剧和文艺片，A 喜欢动作片和喜剧这与 B 的兴趣较相同，A 与 C 都喜欢文艺片，那么如果过我们要预测 A 对一部文艺片的评分时，而与其相似的这两个用户都对该片评分了，那么这两个用户中哪一位的评分最有价值呢？这显而易见，C 很喜欢文艺片，意思是他常看文艺片，他与 A 用户因正是因为这一点而相似的，那么 C 的评分无疑对 A 对这部文艺片的评价有重大的影响，B 用户虽然看了此片，但没有看过此类其他影片，或者说他并不喜欢这类影片，这说明他与 A 用户也不是在这类影片上体现相似性，其参考价值就不高了，甚至会因为其不喜欢这类的影片而打了较低的分，这样如果将其放入产生 A 对这部文艺片的评分预测中计算，因为其相似度最高，势必会大大影响推荐质量。这样的情况是很常见的，因为每个人都有自己主要的兴趣所在，看了哪部电影并不代表该用户就对这一类的电影都有兴趣，因为用户无论看电影或是购物有时是有很大的随机性（朋友劝说或是因为宣传、拍摄的好、有名导演等等多种因素），如何挑选在某类商品上真正有兴趣的用户，这对协同过滤推荐系统的质量的提高无疑起着至关重要的作用。

我们将要提出的算法正是为了解决以上问题。以下我们将作详细介绍。

5.2.2 相关工作

为了要解决传统协同过滤忽略用户在不同类项目上的喜好的差异而导致的推荐质量不高问题，不少研究者使用项目的信息对项目进行分组，如[37][38]文中都提及利用项目类型信息（如动作片、动画片、喜剧等）作为属性，将项目分到不同的组，然后再在每个组中计算用户的相似性，提高了查找最近邻居的准确性，较为充分地利用了项目信息，但大多数的项目往往同时属于不止一个组甚至多个组，需要分别在不同组分别计算，最后再整合结果，而且单单基于不含有相同的类型信息的两个项目就断言两个项目

没有相似的可能性，用户就不会同时选择二者，这有些不准确。就此，文献[37]进一步分析，认为用户在同一组项目中的相似性是变化较小的，而在不同组的变化较大，提出基于用户相似性的变化将项目聚类，姑且不说其理论的基础是否在任何情况下都是对的，但其对项目的聚类的时间复杂度较前面的利用项目本身信息进行的分类大了许多，对项目聚类需要反复计算用户相似度的值，其代价无疑比较高，而且所谓的用户相似度值变化是针对各个用户的相似度变化平均值而言的，没考虑每个用户对项目的归类的个人意见的差异性，缺乏个性化。

5.2.3 改进的基于用户的协同过滤算法

算法的主要思想：以基于用户协同过滤算法为主，由 4.1 节中基于用户协同过滤算法的介绍可知该算法共有三步，其中我们的主要改进在其第二步，即确定的目标用户的最近邻居，在原来的最近邻居的基础上，利用基于项目协同过滤算法的第二步得到的要预测项目的最近邻居，对用户最近邻居进行进一步的筛选，到“真正”的目标用户的最近邻居。

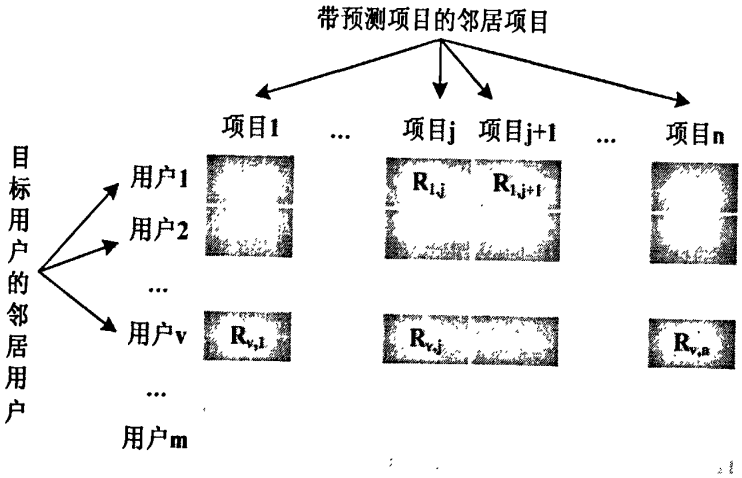

图 5.1 对目标用户的邻居用户再选择

Fig. 5.1 Select neighbors of active user again

如图 5.1，其中用户 1、用户 2 和用户 v 为目标用户的邻居用户，而项目 1、项目j、项目j+1 和项目 n 为与带预测项目相似的项目，对我们的算法而言，关注的是相似项目和邻居用户两个维交叉的地方，即图中颜色最深的部分。我们需要观察的是对于每一个邻居用户是否对待预测项目的相似项目中至少一个做出评分，如果是这样，那么该邻居用户将被保留，否则如果没对任何一个待预测项目的相似项目中的项目评分，我们则认为该用户对待预测项目这类项目不感兴趣或者说没有和用户类似的兴趣，那么他产生目标用户对待预测项目的评分就没有多大意义，而且可能带来负面影响。而对最终产生推荐有意义的是那些被保留的邻居用户，他们对待预测项目这类项目感兴趣或者说他们目

标用户在这类项目上有相似的偏好，这些用户更有价值，也体现了个性化推荐的思想。在图 6.1 中，用户 1 对项目j 和项目j+1 均有评分，用户 v 对项目 1 项目j 项目 n 均有评分，他们都将作为目标用户最终产生推荐的邻居用户，而用户 2 没有对待预测项目的邻居项目中任何一个评分，它将被剔除出邻居用户集合。

为什么我们考虑使用基于项目协同过滤算法中的前两步来找与待评价项目相似的项目呢？这是因为基于项目协同过滤中得到的与待预测项目的邻居集合是用户自己选择的（用户已评分的），可以说着某种程度上是用户对项目的分类方式，反映了用户认为与待预测项目相似的项目，这比宏观意义上根据项目信息和考察用户整体将项目分类（如文献[37][38]）更有意义。

那么基于项目协同过滤得到的项目邻居的个数需要很大吗？如果项目邻居个数很大，那么就不能从用户的邻居集合选择出真正相似的用户，这样就和整体度量用户的相似性差不多了，等同于原始的基于用户的协同过滤算法差不多了，所以项目的邻居个数需要较小，而且需要尽量保证，得到邻居项目与目标项目比较相似，否则就不能确定二者为一类的意义了。

假设我们要预测用户 u 对项目 i 的评分，我们的算法的具体实现过程如下：

输入：用户评分矩阵 R；

输出：用户 u 对项目 i 的评分 $P(u,i)$ 。

第一步，分别计算用户间相似度和项目间相似度，参照 4.1.1 节和 3.1.1 节。

第二步，针对要预测的用户 u 对项目 i 的评分，分别确定基于用户和基于项目的协同过滤算法的相应的最近邻居集合 N_u、N_i，具体方法分别参照 4.1.2、3.1.2 节，其中基于项目的协同过滤算法中的最近邻居集合 N_i 个数不需要太多，因为我们需要与项目 i 比较相似的项目。

第三步，针对基于用户协同过滤算法中得到的用户 u 的最近邻居集合 N_u 中每一个用户 v，通过查询评分矩阵，判断他是否对基于项目的协同过滤算法得到的项目 i 的最近邻居 N_i 中的项目评分，如果一个没有，说明他并不是很喜欢项目 i 这类电影，或者说他与目标用户 u 主要不是在这类电影上具有相似性，我们就将这个用户从 N_u 中剔出，当每一个在邻居用户集中的用户 v 都进行判别后，最后我们得到目标用户 u 的针对项目 i 来而言“真正“的最近邻居集合 N_u'。

最后，利用上一步产生的最近邻居集合 N_u' 产生推荐，推荐方法与 4.1.3 节介绍的基于用户的协同过滤算法相同。

5.2.4 算法分析

通过结合基于用户和基于项目协同过滤算法，弥补了二者各自的不足，以基于用户协同过滤算法为主，维护了角色一致性，保留了基于用户的协同过滤系统容易挖掘出目标用户潜在的新兴趣，即能够实现奇异发现，解决基于项目的协同推荐不能作出“跨类型”的推荐的缺点，而由基于项目协同过滤算法得出的要预测项目的最近邻居集合，提高了基于用户协同过滤算法对于目标用户的最近邻居选择的精确性，从而提高了预测精度，改善推荐质量，一定程度上减轻了由于用户评分矩阵稀疏导致的相似度度量不精确问题对推荐质量的影响。

我们提出的算法的最大的特点是我们反映了用户对不同类的项目对应的相似用户是变化的，而不是传统意义上的完全依赖用户间的对所有项目评分的整体相似性确定与目标用户的相似用户。我们提出的改进的基于用户协同过滤算法通过使用基于项目协同过滤算法得到的目标用户已评价的与要预测的项目相似的项目集合，来对与目标用户整体相似的用户进行进一步选择，得到在项目 i 此类项目上与目标用户相似的邻居用户，以此去产生最终预测评分。

而且我们提出的算法与以往使用项目类型对项目进行分组归类不同，因为归类是总体性的，而不是因人而异的，我们都有这样的经验，因为人们对同一组事物的分类不尽相同，因为每个人的经验和知识程度是不尽相同，而我们的算法体现了这一点，即我们通过基于项目协同过滤找到的与待预测项目相似的项目是用户“认为”的，用户在与待预测项目较为相似的几个项目中评了分的。

上一节我们已经讨论了算法的第二步中，待预测项目的最近邻居的确定无疑对算法的效果影响很大，如何确定待预测项目的最近邻居项目是很关键的步骤，这一点我们将在下文通过实验找出较好的方式。

本算法的计算量似乎增加了许多，其实不然，因为基于项目协同过滤推荐算法大部分工作可以离线计算。因为在典型的电子商务环境中，项目之间的关系相对来说比较稳定，所以利用项目之间的相似性，因此可以离线完成工作量最大的相似性计算步骤，然后根据具体的目标用户和要预测项目通过查找已离线计算完的项目间的相似性而得到项目的最近邻居集合，而且我们的算法只需要很少与要预测的项目的比较相似的项目，这样在这里花费的时间很小，然后根据这个邻居项目集合选择基于用户协同过滤得到的用户的邻居集合。所以我们在线部分只比原来的基于用户协同过滤算法多了要预测项目的最近邻的确定和对用户的最近邻居的再选择这两个耗时很少的简单步骤，而我们算法将呈现的较高的推荐质量以及加强对用户推荐的个性化方面的贡献是明显的，所以算法的添加的部分的时间空间的消耗是可以容忍的。

5.3 实验与分析

5.3.1 实验方案

我们实验中算法涉及到的基于项目和基于用户协同过滤相似度计算和产生推荐的方法均与前两章实验部分使用的相同，即相似度度量均为修正余弦相似性方法，分别参照公式 3.2 和 4.1，我们提出的算法最终推荐方法参照传统基于用户协同过滤产生推荐的方法，见公式 4.3。

我们实验共分为两部分：

(1) 改变基于项目协同过滤算法确定目标项目的最近邻居集合的个数的方式，观察其对我们提出的算法的影响。找到较好的确定最近邻居项目的方式。

方式一：直接使用邻居个数限定，相应的可预测评分的个数

方式二：使用相似度阈值限定，相应的可预测评分的个数

(2) 与传统的基于用户协同过滤算法进行对比实验，观察本章算法的是否改善传统基于用户协同过滤算法的推荐质量。

5.3.2 实验结果

5.3.2.1 基于项目协同过滤得到最近邻居的方式对算法效果的影响

为了保证我们算法的效果，找到与待预测项目较相似的项目，我们使用不同的方式确定待预测项目的最近邻居项目的个数，找到较好的方式。

(1) 方式一：直接使用邻居个数限定，我们的算法推荐效果结果和可预测评分个数分别如图 5.2 和 5.3 所示。

从图 5.2 中我们看到，变化待预测项目的最近邻居集合中项目个数，我们的推荐算法的效果有明显的变化。当项目个数增大时，MAE 的值随之增大，这说明推荐质量降低了，这是因为这时得到的邻居与待预测项目的相似性下降，而且相似项目个数变大，目标用户的邻居用户在其中没有评价一个相似项目的几率也降低了，这样就不能有效地对用户的邻居用户进行选择，我们算法的优越性就减弱了。

同时我们也注意到图 5.3 中，可预测的评分个数（即要预测的评分在限定条件下（如用户最近邻居个数），能预测多少）随着相似项目个数的减小而降低，这是因为目标用户的邻居用户在其中没有评价一个相似项目的几率升高，从而达不到算法限定的用户的邻居数，因而能预测的评分个数减少。我们知道在用户评分矩阵极度稀疏的情况下，不是每个用户都能找出邻近用户，再加上邻居个数的限定，所以即使原始的基于用户协同过滤算法也不可能预测全部，这是可以理解的。所以我们只能尽可能兼顾两方面的需求，我们观察到 Number<=8 与 Number<=10 对预测精度和评分个数的影响较相近，我们选择 Number<=8 为方案中最佳。

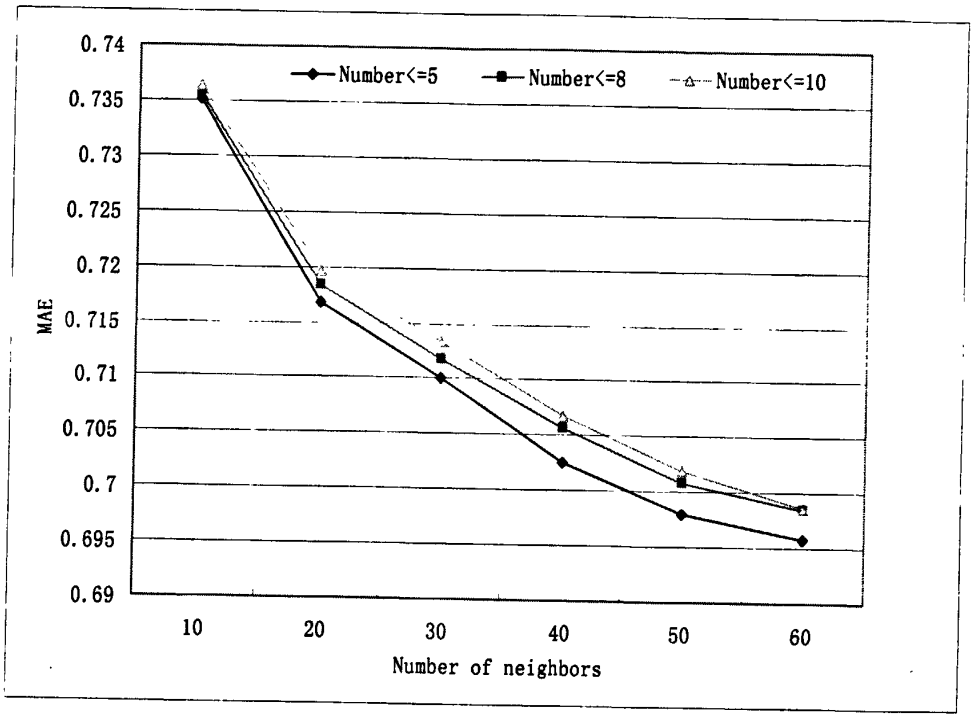

图 5.2 项目邻居个数对 MAE 的影响

Fig. 5.2 The impact of number of item's neighbors on MAE

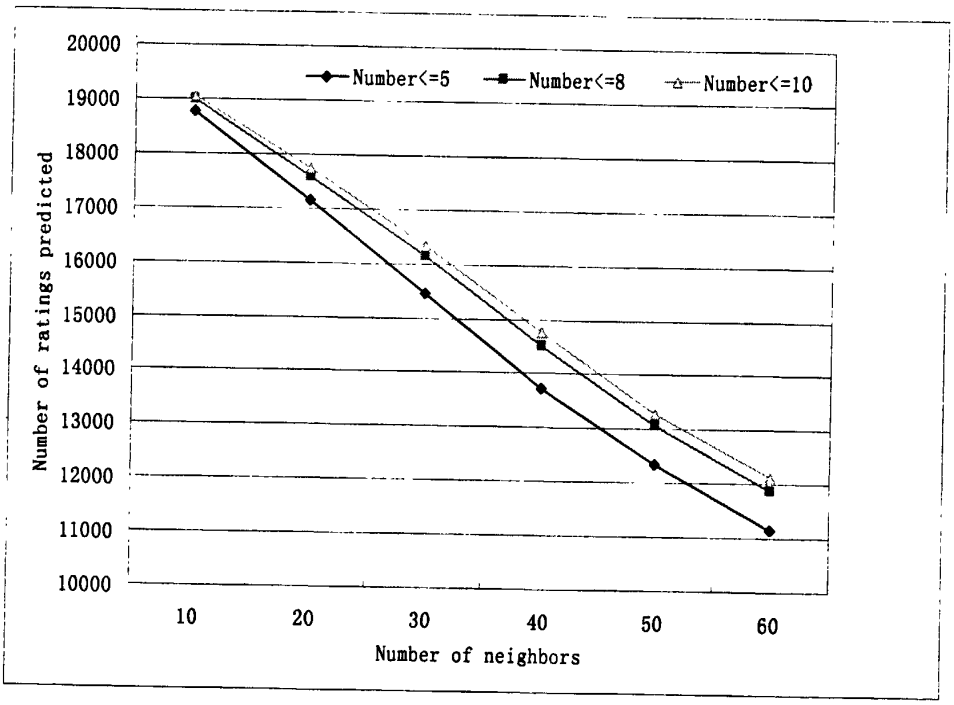

图 5.3 项目邻居个数对预测评分个数的影响

Fig. 5.3 The impact of number of item's neighbors on number of ratings predicted

(2) 方式二：使用相似度阈值限定，我们的算法推荐效果结果和可预测评分个数分别如图 5.4 和 5.5 所示。

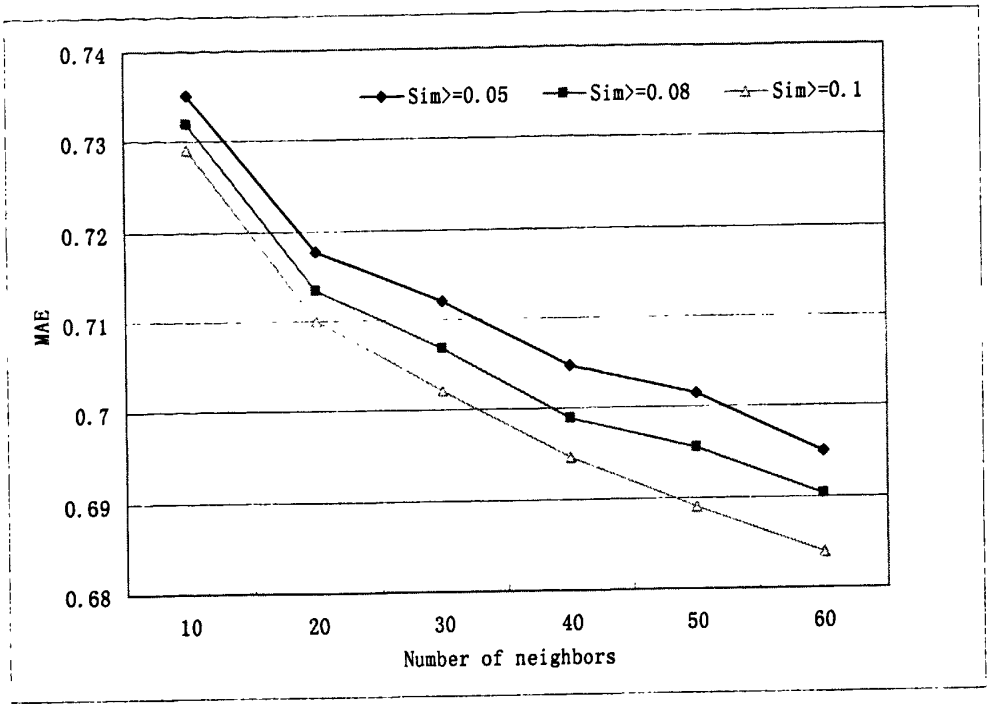

图 5.4 项目相似度阈值对 MAE 的影响

Fig. 5.4 The impact of item similarity' s threshold on MAE

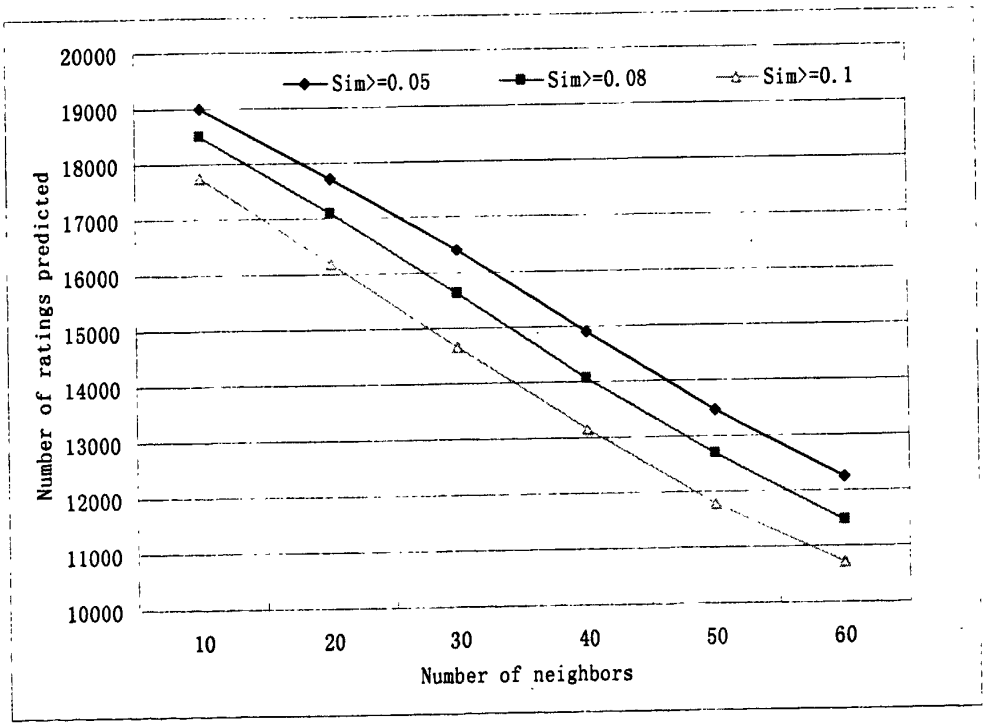

图 5.5 项目相似度阈值对预测评分个数的影响

Fig. 5.5 The impact of item similarity' s threshold on number of ratings predicted

从图 5.4 中，我们看到随着相似度的阈值的增大，MAE 的值也随之降低，这说明推荐质量提高了，这是因为这时得到的邻居与待预测项目的相似性升高，更好的实现对用

户最近邻居的选择，当然我们从图 5.5 中也注意到随着阈值的增加，推荐质量的提高，最近邻居的再选择的严格性也随之提高，导致可预测评分个数的降低，而且降低的较为明显，这里我们选择可预测个数最高的，而其预测精度最差，相似度阈值为 0.05，这与方案一中我们选择的 Number<=8 的可预测评分个数相近。

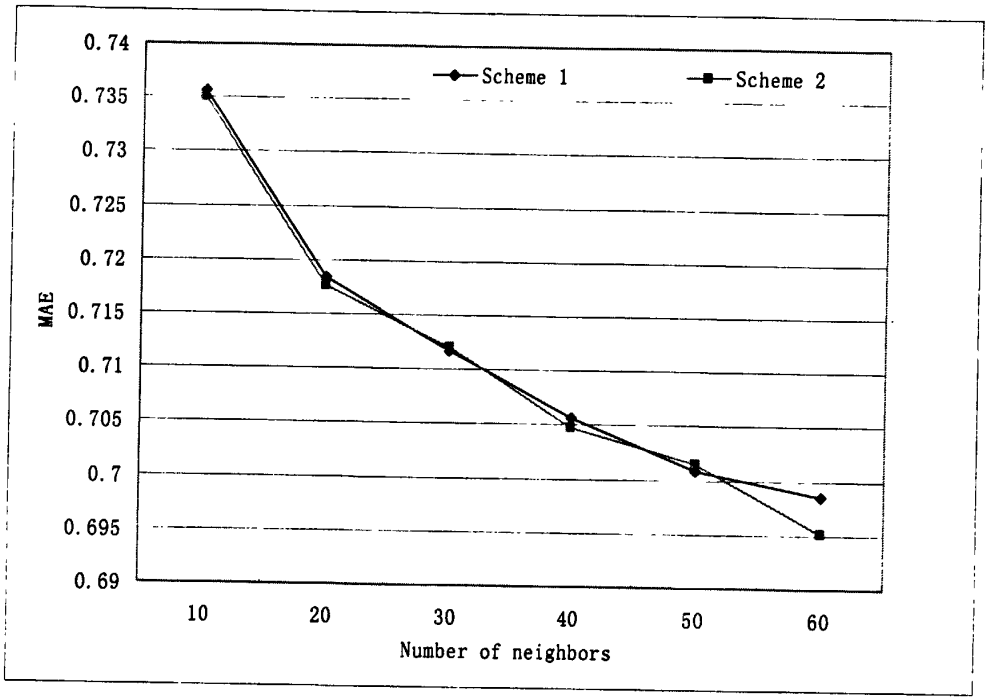

图 5.6 方案一和方案二的 MAE 比较

Fig. 5.6 Comparison of Scheme 1 and Scheme 2 on MAE

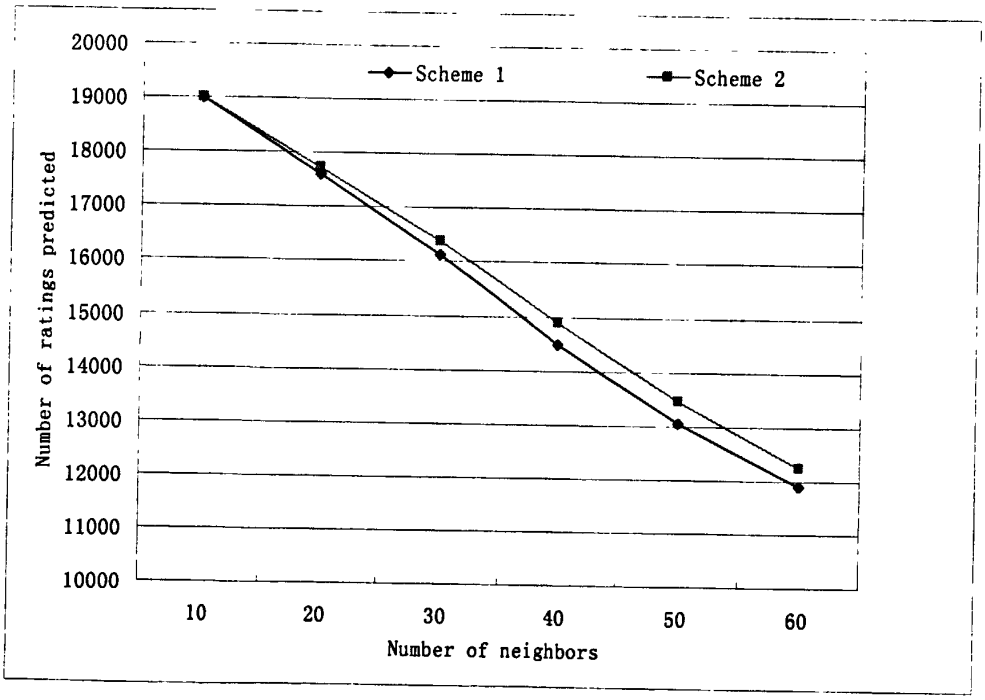

图 5.7 方案一和方案二的可预测评分个数比较

Fig.5.7 Comparison of Scheme 1 and Scheme 2 on No. of ratings predicted

(3) 结合分析

将方案一直接使用邻居个数限定中我们选择的最佳方案与方案二以相似度阈值限定方法中我们选择的二者进行比较，如图 5.6 和 5.7 所示。

由图 5.6 中我们发现方案一和方案二的 MAE 值在邻居个数小于等于 50 时，很接近，几乎是交替为较小，当用户邻居个数为 60 时，方案二比方案一的 MAE 值明显小了许多，但仅凭此图并不能完全确定方案二更好，我们再观察二者可预测评分个数的情况，从图 5.7 中我们发现方案二虽然开始和方案一可预测的评分个数很接近，但随着用户邻居个数的增大，方案一的折线下降的更快，而且除了开始时二者的值几乎相等外，在其他各种情况下，方案二与方案一相比可以预测的评分个数均较多。综合二者在 MAE 和可预测评分个数方面的影响，我们选择方案二。

5.3.2.2 与传统的基于用户得协同过滤算法比较

通过上面的实验，我们的算法选择方案二即相似度阈值限定作为基于项目协同过滤部分确定目标项目的最近邻居的方式，即阈值为 0.05。我们将其与传统基于用户协同过滤算法相比较，如图 5.8 所示。

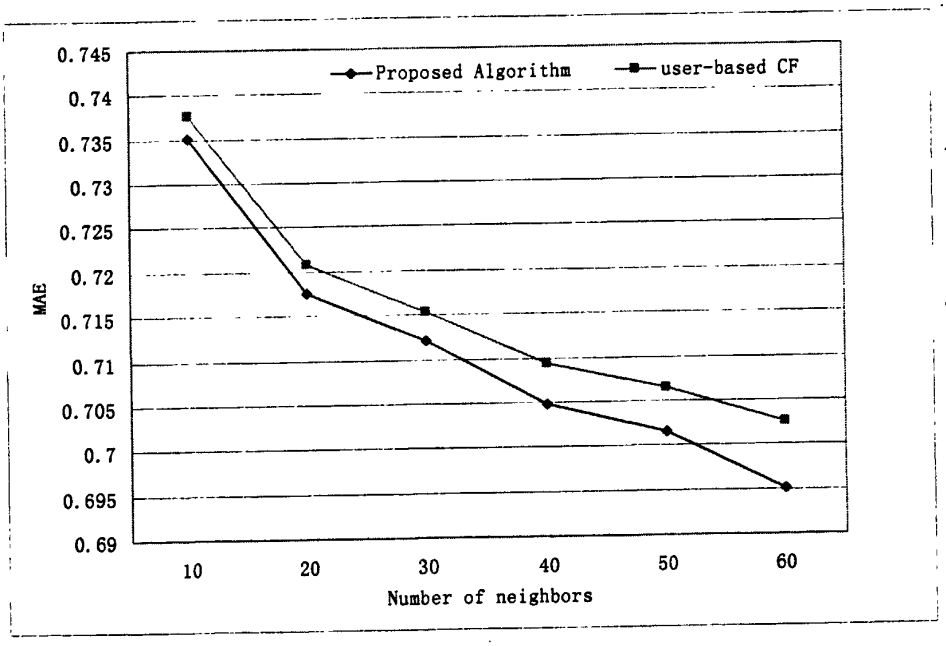

图 5.8 算法准确性比较

Fig.5.8 Comparison of accuracy of algorithms

由图 5.8 我们看到我们提出的算法在不同的邻居用户个数下，均比传统基于项目协同过滤算法的 MAE 值都低，这说明预测精度提高了，而且随着的邻居用户个数的增大，二者间的 MAE 的差距在不断的扩大，这说明当邻居用户与目标用户相似性变小时，我们的算法的对邻居用户的再选择的优势和有效性更加明显，这说明我们的算法对于用户评分矩阵的稀疏性也有一定的适应性，更大程度地提高了推荐系统的质量。

5.3.3 实验结果分析

通过以上一系列实验，我们可以看出改进的基于用户的协同过滤算法取得了很好的推荐效果，而且随着用户邻居个数的增加，其比传统基于用户协同过滤的预测精度好的更多，这是因为这时后加入的邻居用户与目标用户的相似性减小，算法对用户邻居的再选择显得更为必要，因而其效果也随之扩大。而相似性的减小，一方面是由于用户间的评分差异，一方面是由于用户的评分矩阵变得更为稀疏，共同评分项目变得更少，前者是当相似度变化不大时的主要因素，而后者是当相似度急剧下降时的主要因素，而我们的算法当邻居用户个数增加，即相似度减少时，MAE 的值与传统基于用户协同过滤的差值不断扩大，这说明我们的算法对于相似度较小的情况体现了更大的优势，同时也一定程度上改善了由于用户评分矩阵变得稀疏而导致相似性变小的问题。

通过方案一、二的比较，我们注意到当由基于项目协同过滤算法得出的待预测项目的邻居个数较少，相似性较高的情况下，算法的有效性更为明显，预测精度更高。但同时我们也注意到可预测的评分个数也随之减少，这是由于对基于用户协同过滤的用户邻居再选择，自然会减少邻居用户的个数，而再选择的标准越高（即待预测项目的邻居个数越少），能留下的目标用户的邻居用户就越少，因而当我们以用户邻居个数为限定条件来观测 MAE 的值变化时，就会使未达到用户邻居指定个数的待预测评分被丢弃，所以传统基于用户的协同过滤算法也不可避免有这样的情况。这就是一个平衡问题，可以根据实际需求，更侧重于哪一方面，预测精度还是可预测的个数，作相应的调整。

通过以上分析，我们可以预想，如果提高项目间的相似度的计算精度，待预测项目的邻居项目将被更准确的查找，而对目标用户的邻居集合再选择会更为精准，这无疑将给我们的算法提升推荐效果以更大的空间。本文是利用传统的基于项目协同过滤得到待预测相似项目的邻居集合，现在有不少优化基于项目协同过滤的研究成果，可以将其研究成果用于本文算法，以期获得更好的推荐质量。而且是否有更好的方式来得到待预测相似项目的邻居集合呢？这些都是可以进一步研究的问题。还有我们目前提出的算法只是剔出对待预测项目的最近邻居中任何项目做出评分的用户，那么没被剔出的用户，对相似项目评价个数的多少和相似度较高的项目的多少的是不同的，那么如果区别对待这些用户（如对产生推荐的方法中加权重），是否会获得更好的推荐效果，都是值得进一步研究的问题。

5.4 小结

本章通过分析传统的协同过滤推荐算法无法反映用户对不同类的项目的偏好差异，使得推荐时缺少个性，推荐质量不高等问题，利用组合推荐思想，结合基于项目和基于用户二种协同过滤推荐算法而提出了改进的基于用户协同过滤算法。通过进行一系列实

验，证明了与传统基于用户协同过滤算法相比，该算法明显提高了预测评分精度和改善推荐质量，更好地体现推荐系统推荐的个性化。

第六章 结论与展望

6.1 本文主要内容总结

协同过滤作为在电子商务推荐系统中应用较为成功的技术之一，对推荐系统的应用和实施起了很大的推动作用。但随着推荐系统本身及客观环境要求的不断提高，协同过滤面临着许多问题，国内外的学者及研究人员在不断地探索解决这些问题的方法，并取得了一定的成果。论文在深入研究和比较各种方法的基础上，重点对协同过滤技术中的存在的问题及问题出现的原因进行了分析，并在此基础上提出了解决这些问题的新方法。

本文通过对电子商务推荐系统中的协同过滤推荐技术以及其存在的问题等的研究，分析电子商务推荐中的实际问题与需求，针对协同过滤推荐算法的一些缺陷，并结合当前个性化服务推荐技术的前沿，主要内容有：

在传统协同过滤基于用户评分矩阵的计算的基础上，考虑到用户评分数据稀疏性和项目分类信息对项目以及对用户相似性的影响，引入项目类型信息参与传统协同过滤中相似度的计算。本文采用线性结合方式将通过 用户评分矩阵计算的传统相似度和通过项目类型信息计算的相似度结合一并作为事物的相似度。本文分别就此想法在基于项目协同过滤和基于用户协同过滤中的具体的实现作了说明，并进行了相应的实验。实验表明采用本文提出的方法计算相似性，它分别在基于项目和基于用户的协同过滤算法中的应用对推荐的质量都有不同程度的改善和提高。

本文还就传统的协同过滤推荐算法无法反映用户对不同类的项目的偏好差异，使得推荐时缺少个性，推荐质量不高等问题，提出了利用组合推荐思想，结合基于项目和基于用户二种协同过滤推荐算法的改进的基于用户协同过滤算法。该算法利用基于项目协同过滤产生的待预测项目的最近邻居集合对基于用户协同过滤产生的目标用户的最近邻居用户进行再选择，淘汰没有对待预测项目的最近邻居集合中任何项目评分的用户，针对具体每一个项目，找到真正与目标用户相似的用户，通过进行一系列实验证明算法提高预测评分精度和改善推荐质量，进而更好地体现推荐系统推荐的个性化。

6.2 未来工作

本文提出的项目类型优化在基于用户协同过滤算法中的使用不是很理想，这里是否有进一步改进的可能，而且由于用户类型信息对用户相似度影响的间接性，由此我们想到，如果我们也可以从用户信息中找到（如用户分类）的信息加以利用，使用该信息直接作用于基于用户的协同过滤算法的相似度计算，那么我们可以预期能得到更为显著的

影响，我们大胆地期待这也能一定程度上提高推荐质量，同时我们亦可试图将这一信息通过用户评分数据进行转换，在作用于基于项目协同过滤算法，提高推荐质量，因为什么样的项目会被什么样的用户所喜欢，这个信息对推荐无疑是有着重大意义的，通过利用用户信息，可以更直观地观察喜欢某项目的用户的共同特征，加强其在计算相似度中的比例，自然会提高计算项目相似度的精确度，从而改善推荐质量。这些都是下一步需要研究的课题。

本文提出的改进的基于用户协同过滤算法还有进一步研究和优化的可能，如，本文只是剔出没有对待预测项目的最近邻居中任何项目做出评分的用户，那么没被剔出的用户，对相似项目评价的多少应该有所区别，适度的调整相似度，还有本文是利用传统的基于项目协同过滤得到待预测相似项目的邻居集合，现在有不少优化基于项目协同过滤的研究成果，可以将其研究成果用于本文算法，以期获得更好的推荐质量。而且是否有更好的方式来得到对预测相似项目的邻居集合呢？这些都是可以进一步研究的问题。

参考文献

1. 何安.协同过滤技术在电子商务推荐系统中的应用研究[D]，浙江大学，2007.
2. 张光卫，李德毅，李鹏等.基于云模型的协同过滤算法[J]，软件学报，2007，18(10)：2403-2411.
3. Sarwar B，Karypis G，Konstan J，Riedl J．Analysis of recommendation algorithms for E-commerce [A]. In: ACM Conference on Electronic Commerce [C], 2000, 158-167.
4. Mobasher B, Dai H, Luo T, Sun Y and Zhou, J. Integrating Web Usage and Content Mining for More Effective Personalization[A]. In Proceedings of the International Conference on E-Commerce and Web Technologies (ECWeb2000) [C], 2000, 165-176.
5. Mobasher B, Dai H, Luo T, and Nakagawa M. Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data [A]. In Proceedings of the UCAI 2001 workshop on Intelligent Techniques for Web Personalization (ITWPOI) [C], 2001, 53-60.
6. Schafer J B, Konstan J A and Riedl J. Recommender Systems in E-Commerce [A]. In ACM Conference on Electronic Commerce (EC99) [C], 1999, 158-166.
7. Schafer J B, Konstan J A and Riedl J. E-commerce recommendation application [J]. Data Mining and Knowledge Discovery, 2001, 5(1-2): 115-153.
8. Zan H, Hsinchun C, Daniel Z. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering [J], ACM Trans. on Information Systems, 2004, 22(1):116-142.
9. Thiesson B, Meek C, Chickering D, Heckerman D. Learning mixtures of DAG models[A]. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence[C], 1998, 504-513.
10. Aggarwal CC, Wolf J, Wu KL, Yu PS. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering [A]. In: Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C], 1999, 201-212.
11. Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms [A]. In: Proc. of the 10th International World Wide Web Conf.[C], 2001, 285-295.
12. Chickering D, Hecherman D. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables [J]. Machine Learning, 1997, 29(2/3): 181-212.
13. Adomavicius G, Tuzhilin A. Expert-Driven Validation of Rule-Based User Models in Personalization Applications [J], Data Mining and Knowledge Discovery, 2001, 5(1-2): 33-58.
14. Han J W, Kamber M. 数据挖据：概念与技术[M]，北京：机械工业出版

社,2007,251-251.

15. Li Q, Zhou M. Research and design of an efficient collaborative filtering predication algorithm[A]. In Parallel and Distributed Computing, Applications and Technologies[C], 2003, 171-174.
16. Chee S H S, Han J and Wang K. RecTree: A Linear Collaborative Filtering Algorithm [D], Simon Fraser University, 2000.
17. Chee S H S, Han J and Wang K. RecTree: A Linear Collaborative Filtering Method [A]. Proceedings of Data Warehouse and Knowledge Discovery[C], Munich, Germany, 2001, 141-151.
18. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference[M]. San Francisco: Morgan Kaufmann, 1988.
19. 简育华. 基于贝叶斯网络的一种常规雷达目标识别方法[J], 科学技术与工程, 2007, 7(2): 230-235.
20. 孙小华. 协同过滤系统的稀疏性与冷启动问题研究[D], 浙江大学, 2005.
21. Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[A]. In: Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence (UAI'98)[C], 1998, 43-52.
22. Goldberg D, Nichols D, Oki BM, Terry D. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.
23. Resnick P, Iacovou N, Suchak M, et al. Grouplens: An open architecture for collaborative filtering of netnews [A]. In: Proc. of the ACM CSCW'94 Conf. on Computer-Supported Cooperative Work[C], 1994, 175-186.
24. Shardanand U, Maes P. Social information filtering: Algorithms for automating "Word of Mouth"[A]. In: Proc. of the ACM CHI'95 Conf. on Human Factors in Computing Systems[C], 1995, 210-217.
25. Hill W, Stead L, Rosenstein M, Furnas G. Recommending and evaluating choices in a virtual community of use [A]. In: Proc. of the CHI'95[C], 1995, 194-201.
26. Ungar L H, Froster D P. Clustering methods for collaborative filtering [A]. Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence[C], 1998,112-125.
27. 王霞. 协同过滤在电子商务推荐系统中的应用研究[D], 海河大学, 2003.
28. MARKO B, YOAV S. FAB: content-based collaborative recommendation [J]. Communications of the ACM, 1997, 40(3): 66-72.
29. Sarwar B, Karypis G, Konstan J, et al. Application of Dimensionality Reduction in Recommender System-A case study [A]. In: Proc. of the ACM WebKDD 2000 Workshop[C], 2000, 82-90.
30. Deerwester S, Dumais S T ,Furnas G W, Landauer T K and Harshman R. Indexing by

Latent Semantic Analysis[J],Journal of the American Society for Information Science,41(6): 391-407,1990

31. Golub G H and Van Loan C F. Matrix Computations(Third Edition)[M], The Johns Hopkins University Press, 1996
32. 周云辉. 电子商务个性化推荐技术及其应用研究[D]，北京邮电大学，2004.
33. Aggarwal CC. On the effects of dimensionality reduction on high dimensional similarity search [A]. In: Proc. of the ACM PODS Conf.[C], 2001,256-266.
34. Yu K, Wen Z, Xu X, et al. Feature Weighting and Instance Selection for Collaborative Filtering [A]. 2nd International Workshop on Management of Information on the Web, in conjunction with the 12th International Conference on DEXA'2001[C], 2001.285-290.
35. Yu K, Au X W, Tao J H, et al. Instance selection techniques for memory-based collaborative filtering [A]. In Proceedings of the second international conf. on data mining[C], 2002, 59-74.
36. 周军锋，汤显，郭景峰. 一种优化的协同过滤算法[J]，计算机研究与发展， 2004，41(10)：1842-1847.
37. Quan T, Fuyuki I, Shinichi H. Improving accuracy of recommender system by clustering items based on stability of user similarity [A]. IAWTIC'2006 Proc [C], 2006, 61--68.
38. 张卫光，康建初，李鹤松等.面向场景的协同过滤推荐算法[J]，系统仿真学报增刊，2006，18(2)：595-601.
39. ROBIN B. Hybrid recommender systems: survey and experiments [J]. User Modeling and User Adapted Interaction, 2002, 12(4): 331-370.
40. 邓爱林，朱扬勇，施伯乐.基于项目评分预测的协同过滤算法[J]，软件学报,2003,14(9)：1621-1628.
41. 王惠敏，聂规划. 融合用户和项目相关信息的协同过滤算法研究[J]. 武汉理工大学学报，2007，29(7)：160-163.
42. Ma H, King I, Lyu M. Effective missing data prediction for collaborative filtering [A]. In: Proc. of the 30th annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR) [C], 2007, 39-46.

致 谢

光阴似箭，转眼间为期两年的研究生学习生活即将结束。很庆幸自己能够来到计算机软件理论研究所这个充满朝气、团结向上的集体。在软件所度过的日日夜夜里，我无时无刻不被老师们严谨求实的治学作风和同学们刻苦钻研的学习态度所感染，这种经历不仅使我在学业上不断进步，而且在生活的诸多其它方面也受益非浅。

首先要感谢的是我的导师王大玲教授。王老师以敏锐的眼光、饱满的智慧、渊博的知识，卓越的才华站在学术的前沿，作为王老师的学生我深感骄傲和自豪。在研究生期间，王老师始终给予我严格的要求、充分的信任、热情的鼓励和全面锻炼的诸多机会。学术研究中，我深受王老师渊博的学识、开阔活跃的思维、对学术前沿的敏锐触感和准确把握、严谨的治学态度和勤奋的敬业精神所熏陶和激励，并且从中获益良多；在生活中，王老师也同样给予了我无微不至的关怀和照顾。在此，我对恩师表示最崇高的敬意和最诚挚的感谢！

感谢于戈老师给予我的指导和帮助，他正直的为人、踏实细致的工作作风、精辟的学术观点和广博的学术知识令我印象深刻，这些将使我受益终生。

感谢鲍玉斌老师，鲍老师精益求精的态度、忘我的工作精神和平易近人的待人方式对我产生重要的影响。我衷心的感谢鲍老师给予我的无私的教诲与帮助。

同时还要感谢申德荣老师、董晓梅老师、杨晓春老师、林树宽老师、邓庆绪老师等其他软件所的老师在学习和生活中对我的帮助和关心。

感谢海洋组的冷芳玲老师、宋杰师兄的指导与关怀。

感谢海洋组的其他成员和数据仓库与数据挖掘小组的全体成员，他们都在学习中给予了我很大的帮助。

感谢我的父母这么多年对我的养育之恩，他们克服了各种困难在学习和生活的各个方面都给予了我最大的支持和鼓励，父母的关爱是我能够坚持走完漫长求学道路的最大动力。

再次感谢所有关心和帮助过我的老师和同学们。祝愿软件理论研究所的明天更加辉煌。

攻硕期间参加的项目和发表的论文

参加项目：

1. 国家海洋908重点专项课题“海洋数据体系规划和海洋数据仓库构建技术”（项目编号：908-03-06-01），2007.1~2008.12，110万元，项目主要参与人
2. 国家自然科学基金项目“面向新一代搜索引擎的用户动机推演模型的研究”（编号：60573090），2006.1~2008.12