中图分类号：

TP311.1：

负相关性信息和Sigmoid权重相似度对协同过滤算法的影响研究

管理科学与工程

周继平

郭强 副教授

二〇一三年十二月

学校代码：10252
学　　号：112480748

上海理工大学硕士学位论文

负相关性信息和 Sigmoid 权重相似度对协同过滤算法的影响研究

姓　　名　　周继平
系　　别　　管理学院
专　　业　　管理科学与工程
研究方向　　信息管理与决策支持系统
指导教师　　郭强 副教授

学位论文完成日期　　2013 年 12 月

University of Shanghai for Science and Technology

Master Dissertation

The Impact of Negative Correlation and Sigmoid Weight Similarity on Collaborative Filtering Algorithm

Name	Zhou Jiping
Department	Business School
Specialty	Management Science and Engineering
Research Direction	Information Management and Decision Support System
Supervisor	Associate Professor Guo Qiang

Complete Date December 2013

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学位论文保留并向国家有关部门或机构送交论文的复印件和电子版。允许论文被查阅和借阅。本人授权上海理工大学可以将本学位论文的全部内容或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于　　保 密 ____ 年　□
　　不保密　　□

学位论文作者签名：　　指导教师签名：

年　月　日　　年　月　日

声　明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下,独立进行研究工作所取得的成果。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

摘　要

互联网的快速发展推动了 Web2.0 时代的到来，网络用户由 Web1.0 时代信息被动接受者变为主动发布者。互联网的普及，促使了网民数量的快速增长，从而带动了互联网信息的爆炸式增长，信息越来越多，然而用户对信息的利用率却反而降低，出现了信息过载和信息迷航，用户从海量内容中找到自己感兴趣的信息所需要的时间成本越来越高。作为解决信息过载问题的工具之一，推荐系统能够根据用户的访问行为记录，挖掘用户兴趣，为每个用户推荐其感兴趣的信息，实现个性化的精准推荐。协同过滤算法是工业界使用最广泛运用最成功的一种个性化推荐算法，但其在应用中遇到稀疏性问题和冷启动问题，极大地降低了推荐系统的精度，阻碍了推荐系统的发展。本文在经典协同过滤算法的基础上，提出了考虑负相关性信息的协同过滤算法和基于 Sigmoid 权重相似度的协同过滤算法，实验证明了新算法的有效性。本文具体的研究工作如下：

1、提出了考虑负相关性信息的协同过滤算法。邻居选择在协同过滤算法中起到承上启下的作用。经典协同过滤算法通常采用 Pearson 相关系数计算相似度，然后选择相似度最高的若干用户作为当前用户的最近邻居，此种算法仅考虑了用户评分的正相关性信息，却忽视了负相关性信息。MovieLens 数据集上的对比实验表明，负相关性信息不仅可以提高推荐结果的准确性还可以增加推荐列表的多样性。此外，我们还发现负相关性信息有助于提高度小用户的推荐准确性。综上所述，负相关性信息有助于解决推荐系统中同时保证推荐列表的准确性和多样性的问题以及冷启动问题。

2、提出了基于 Sigmoid 权重相似度的协同过滤算法。相似度计算是协同过滤算法的基础，邻居选择和评分预测均需要准确的相似性度量。经典协同过滤算法在共同评分的项目上计算相似度，但没有考虑共同评分项目集的大小，后来改进的权重相似度虽然考虑到了这一点，但是仅降低了在较小共同评分集上的相似度，没有增加较大共同评分项目集上的相似度，并且引入了需要手动调节的权重参数。MovieLens 数据集上的实验表明，基于 Sigmoid 权重相似度的协同过滤算法不仅能获得比传统协同过滤算法更好的预测准确性和推荐覆盖率，而且能弥补权重相似度需要手动调节参数的不足。此外我们还发现，该算法能大幅度提高度小用户的预测准确性。综上所述，Sigmoid 权重相似度能有效缓解协同过滤算法中的稀疏性问题和冷启动问题。

上述的研究工作，从一定程度上解决了协同过滤算法所面临的稀疏性问题、冷启动问题以及同时保证推荐列表多样性和准确性的问题，有助于推动协同过滤算法的理论研究和现实应用。

关键词：个性化推荐系统 推荐算法 协同过滤 评分预测 负相关性信息 权重相似度

ABSTRACT

The rapid development of the Internet has promoted the coming of Web2.0 era. Web users have changed from passive recipients of information in the era of Web1.0 to active publisher of information in Web2.0. Popularity of the Internet has prompted a rapid growth of the number of Internet users, thus laded to the explosion of the Internet information, resulting in more and more information. However the utilization of information but instead reduces, information overload and information trek appears. The time that users spend finding their interesting information from mass information is becoming more and more. As one of early tools to overcome information overload problem, recommended system can mine users' interest according to their network behaviors, then recommend the interesting information for each user and make personalized precise recommendation. Collaborative filtering algorithm is one of the personalized recommendation algorithms of recommendation system that most widely and successfully used in industry up to now. But the sparseness problems and cold start problems are always troubling the collaborative filtering. In this paper, a Collaborative filtering algorithm by considering negative correlations information and that based on Sigmoid weight similarity have proposed based on the classic collaborative filtering algorithm for the data sparsity and cold start problems. Their effectiveness is verified through several specific experimental. The following is the corresponding theoretical research and application of the paper for collaborative filtering algorithm:

1、**Firstly, a collaborative filtering algorithm by considering negative correlation information is presented**. Neighbor selection plays a connecting role to collaborative filtering algorithms. Classic collaborative filtering algorithms usually calculates similarity using the Pearson correlation coefficient and selects a few users of the highest similarity as the current user's nearest neighbor, only considering positive correlation information of the user ratings for item, ignoring the negative correlation information. Experiments on MovieLens datasets show that negative correlation information can not only improve the accuracy of the prediction results also increase the diversity of recommendation list. Further analysis reveals that negative correlation information can greatly improve the recommended accuracy of users with small degree. To sum up, negative correlation information helps to solve the dilemma of the accuracy and diversity of recommendation list and cold start problems in recommendation system.

2、**Secondly, a collaborative filtering algorithm based on sigmoid weight similarity is proposed**. Similarity calculation is the basis of collaborative filtering algorithms. Neighbors selection and rating prediction both require accurate similarity. Similarity is calculated on co-rating items in classic collaborative filtering algorithm, but not considering the size of co-rating item sets. In spite of taking into account of it later improved weight similarity only reduces similarity from a small co-rating set, not increase similarity from larger co-rating set. Experiments on MovieLens datasets show that the algorithm can get better performance than the traditional collaborative filtering algorithm on the prediction accuracy and recommendation coverage and compensate for the lack of weight similarity for manually adjusted parameters. Further analysis showed that the algorithm can improve predictive accuracy for users with small degree. In conclusion, sigmoid weights similarity can effectively alleviate the data sparsity and cold start problems in recommendation system.

The above research work solves the data sparseness, cold start problems and dilemma of the accuracy and diversity of collaborative filtering algorithm from a certain extent, thus help to promote collaborative filtering theoretical research and practical application of collaborative filtering algorithm.

Key words: personalized recommend system, recommendation algorithm, collaborative filtering, rating prediction, negative correlation information, weight similarity

目　录

第一章 绪论

1.1 研究背景

计算机软硬件和互联网的迅猛发展，给人们工作和生活的方方面面都带来极大的影响。我们可以在当当和京东商城买到想要的图书和电子产品；在淘宝上买到心仪的衣服；在 1 号店买到自己的日常生活用品；当我们需要去一个陌生的地方时，可以借助百度地图和 Google 地图查找最快最省时的路线；通过 VPN，我们可以实现远程办公等。互联网已成为我们获取信息的主要途径，渗入了我们生活的各个方面。随着互联网高速发展，我们从 Web1.0 时代信息被动接收浏览者变为了 Web2.0 时代中的信息主动创造者，每个人可以不受时间和地域的限制，发布自己的观点。互联网上的信息变得越来越多。据统计，谷歌搜索平均一秒的用户使用量为 200 万，Twitter 平均每天的推特发布量为 3.4 亿条，WordPress 平均每分钟的博客帖子发布量为 350 个[1]。图 1-1 显示了互联网每分钟产生的信息量。

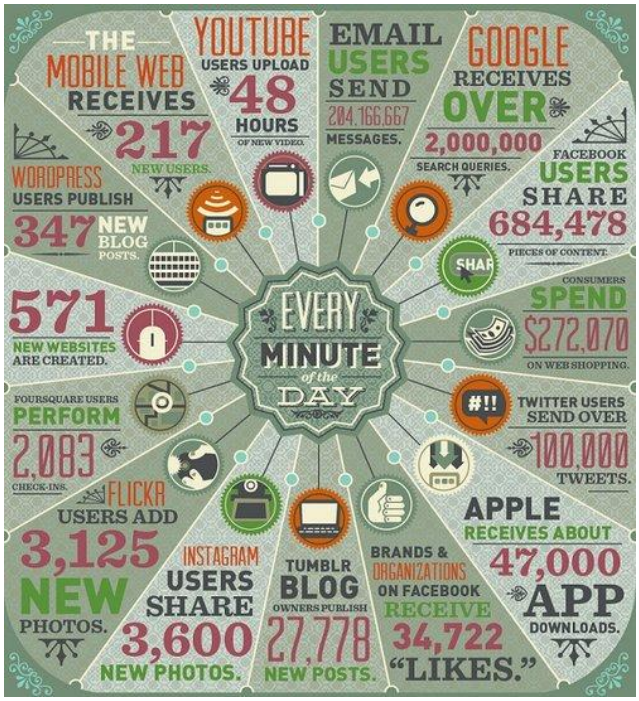


图 1-1 互联网每分钟产生的数据量

海量信息在满足了人们信息需求的同时，也带来了“信息过载”问题。信息过载是指互联网上信息的大幅增长，用户无法从海量信息中找到自己真正想要的信息，出现信息的数量在增长但是信息的利用率反而降低的不合理现象。可见信息过载增加了人们寻找信息的成本，降低了学习工作的效率。为了解决信息过载带来的烦恼，科学家和工程师们提出了很多著名的解决方案，比如搜索引擎。以

搜索引擎为代表的信息检索技术，能帮助用户获取网络信息，一方面，搜索引擎利用关键词采用某种算法对信息进行过滤，将与用户查询需求相关的信息展示给用户。当用户使用同一关键词进行检索时，所获得的结果是一样的，因而无法满足网络环境下用户需求的多元化和个性化。另一方面，搜索引擎完全由用户主导，用户主动输入关键词进行检索，当对搜索结果不满意时，会主动调整搜索关键词重新进行搜索。在这种情况下，如果用户对自己的查询需求不是很明确，那么搜索引擎是无法满足用户需求。由此可见，搜索引擎还没有很好地解决信息过载问题。

在这种情况下，针对信息过载的另一项伟大的技术——个性化推荐系统应运而生。个性化推荐系统能为每个用户提供不同的有针对性的信息服务，它通过分析用户的行为，建立兴趣模型，获得用户的兴趣爱好，为用户提供个性化的信息服务来解决信息过载问题。与搜索引擎相比，个性化推荐系统既能满足用户个性化的需求，而且由于采用推送信息的方式，无需用户提供明确的信息需求。

个性化推荐系统通常由如下几部分组成：收集用户行为的记录模块、分析用户兴趣的分析模块和推荐算法模块，其中推荐算法模块是推荐系统最关键的部分，能决定推荐系统效果的好坏[2]。协同过滤算法是目前推荐系统中使用最为成功和广泛的个性化推荐算法，其基本思想是将与当前用户兴趣相似的其他用户喜欢的项目推荐给当前用户，其最大的优点是对推荐对象没有限制，能处理电影、音乐等非结构化的对象。但由于协同过滤算法本身的特点，随着互联网规模的快速增长，推荐系统中的用户数和项目数迅速扩大，传统协同过滤算法遇到了进一步发展的瓶颈问题，如稀疏性问题[3]、冷启动问题[3]和推荐结果同时保证准确性和多样性的问题[4]等。因此对协同过滤算法进行研究，对进一步推广协同过滤算法的应用范围，推动和促进推荐系统的发展，具有重要的理论研究价值和现实应用意义。

1.2 研究意义

1.2.1 理论意义

Resnick 和 Varian [5]在 1997 年给出推荐系统的定义后，推荐系统逐渐发展成为一个独立的研究领域，特别是协同过滤算法的提出，极大地促进了推荐系统地发展，针对推荐系统的研究成为了信息科学的一个研究热点。

从理论方面来看，推荐系统实际上是一个基于大数据分析技术的高级智能决策支持系统。通过记录用户的行为，如日志、浏览路径、点击和打分等，建立用户兴趣模型，利用大数据分析技术，分析用户的兴趣爱好，从而向用户推荐其感兴趣的物品，既能满足用户的个性化需求，又能为其购物提供决策支持，极大地

改善了用户的购物体验。随着微博、微信和易信等新型信息发布方式的诞生，物联网和移动互联网等新兴技术的迅猛发展，数据正在以前所未有的速度不停地产生，不断地积累，大数据时代已经到来。《Nature》早在 2008 年就设立了 Big Data 专刊[6]，《Science》2011 年推出了专刊“Dealing with Big Data”[7]。国内核心期刊计算机研究与发展也建立了大数据专刊，其中大数据分析专栏下就有两篇关于推荐系统方面的文章[8,9]。由此可见，根据用户行为分析用户喜好从而向其推荐物品的推荐系统正是大数据分析的典型应用，因此个性化推荐系统为大数据技术的研究提供了广阔的舞台。

信息推荐问题是信息挖掘和信息过滤这一科学问题的重要组成部分，涉及到信息科学、管理科学、数学、计算机科学和运筹学等多门学科，是典型的交叉跨领域学科，因此对信息推荐问题的研究，有助于多个学科间知识的融合和创新。

评分预测和 TOP-N 推荐是推荐系统两个主要分支，前者预测出用户如果观看某部电影将会打多少分，后者生成一个由用户可能喜欢的物品组成的推荐列表。基于评分预测的推荐其实就是根据用户评分矩阵中已有的值预测出剩下的缺失值。因此对于推荐系统的研究，有助于解决现代信息科学的中心问题之一：如何从极强噪音的稀疏关联矩阵中挖掘有用的信息[2]。

最近邻模型和矩阵分解是协同过滤算法中的两个最主要的经典模型。在稀疏的用户项目评分矩阵中，矩阵分解可以取得较最近邻模型更好地效果，但是如果用户评分矩阵稀疏度较大，评分矩阵数据缺失严重，那么矩阵分解会产生较大误差，因此对推荐系统的研究，有助于解决数据缺损下的低秩逼近这一困难而复杂的研究课题[10]。

1.2.2 现实意义

从现实方面来看，互联网行业是一个快速发展，竞争激烈的行业，各大电子商务网站都在想尽千方百计获得新的利润增长点。推荐系统不仅可以帮助电子商务网站实现个性化营销，还可以作为客户关系管理的重要组成部分，使每个用户都有不同的商品展示平台和购物体验，大大提高用户对网站的黏着性，从而提升网站的用户点击量。具体来说，推荐系统对电子商务网站的影响包括[11]：

（1）促使电子商务网站用户从浏览者到购买者的转变；

（2）让用户购买原本没有打算购买的商品；

（3）增加用户对电子商务网站的黏性。

电子商务是推荐系统运用最成功的一个领域。商家可以分析用户的购买行为，获得其兴趣爱好，从而将顾客感兴趣的商品推荐给该用户，比如图书、音乐和电影等。目前各大电子商务网站均开始广泛使用个性化推荐系统来为用户推荐商品，

比如淘宝、当当、京东和 Amazon 等，其中 Amazon 被称为是推荐之王，使用推荐系统长达十多年，据 VentureBeat 统计，Amazon 的推荐系统为其贡献了 35% 的商品销售额。

1.3 个性化推荐系统的应用与研究

1.3.1 个性化推荐系统的应用

推荐系统自从 20 世纪 90 年代诞生以来，不论是在应用还是研究领域，都得到了飞速发展。实践证明，推荐系统能够很好地解决信息过载问题，因此很多互联网公司抓住机遇，大力部署和开发推荐系统，并将推荐系统与自身的业务紧密结合。

Amazon 是推荐系统早期应用者，其 35%的商品销售来自推荐系统，素有推荐之王的美称。Amazon 为用户提供个性化的推荐列表和相关商品推荐。个性化推荐列表将与用户之前喜欢的商品相似的商品推荐给目标用户，此外还可能会利用用户的社会关系，将用户的朋友喜欢的商品推荐给目标用户。相关商品推荐将与目标用户订单中相关联的商品推荐给目标用户，这种方式既能打包销售，又能为推荐提供解释，告诉目标用户该组商品有多少用户会一起购买，增加了用户对推荐结果的可信性和推荐系统的透明性，让用户更加信任推荐系统所给出的推荐结果，增加了用户对推荐系统的依赖性和对网站的黏着性。

Netflix 是美国的一家视频租赁服务公司。据统计，其 60%以上的视频订阅租金来自个性化推荐系统，因此非常重视推荐系统的研究与应用，特别是 2006 年 Netflix 举办的推荐系统大赛，悬赏 100 万美元，以将其推荐系统的预测准确性提高 10%。该比赛对学术界和工业界都产生了较大的影响，既为学术界提供了可供研究的公开数据集，又吸引了大批科学家投身到推荐系统的研究中来。此外该比赛产生的很多推荐算法，极大地提高了业界推荐系统的性能。总之，该比赛大大提高了推荐系统在学术界和工业界的影响力。

推荐系统在国外的迅猛发展以及其所拥有的巨大商业和学术价值，很快引起了国内互联网界的重视，在国内的应用范围越来越广，不仅出现了像百度这样构建自己推荐引擎的公司还出现了像百分点这样为他人提供第三方个性化推荐服务的科技公司。

百度从 2011 年试水个性化推荐，按照“百度新首页，一人一世界”的理念，首先推出的就是百度的个性化首页。个性化首页将用户近期以来经常访问关注的网址做成目录的形式，方便用户点击。经过近两年的发展，百度的个性化推荐产品的种类快速增长，形成了自己的个性产品族，如百度视频随心看，百度音乐随身听，百度新闻新版客户端，百度知道推荐等产品。利用流式计算和实时索引技

术，百度推荐系统的更新周期快速缩短，从 20 小时一直缩短到 5 秒钟，提速 14400 倍，用户的转化率因此提高了 3 倍。百度构建的不仅仅是推荐系统，而是建立跨领域的多媒体推荐引擎，最大限度复用数据、算法和系统资源，同时为冷启动问题提供了新的解决方法。业内专家表示百度的推荐引擎技术实力已经达到世界领先水平。

北京百分点信息科技有限公司是一家高科技公司，主要产品包括百分点推荐引擎、百分点分析引擎和百分点个性化 EDM。目前百分点已经和多家知名的互联网公司成为合作伙伴，为他们提供个性化服务等站内流量转化和高级商务智能服务，比如凡客诚品、唯品会和库巴网等。百分点领先的个性化推荐技术能够帮助人们获取最准确和最个性化的信息。上一代互联网实现的是传统行业的整合，将线下的流程转移到线上去，比如购物、求职、求医、交友、相亲和在线教育等，而下一代互联网的趋势有两个，一个就是对人们生活方式的改变，另一个就是效率的提高。百分点这样的互联网公司，整合的不再是传统行业，而是互联网行业，致力于提供互联网行业的效率，因此百分点的产品与下一代互联网的发展方向是相吻合的。

安居客是一家移动互联网公司，专著于房地产行业，是国内最大的房地产信息平台，致力于为各种用户提供最佳找房体验，最终帮用户实现家的梦想。安居客的主要业务包括新房、二手房、租房和商业地产，主要产品包括安居客网站、手机 APP 和 PAD。安居客现有的推荐系统已经全部覆盖了其主要业务和主要产品，其全站流量中有 20%是来自推荐系统。安居客根据用户的行为日志，建立用户兴趣图谱，给用户推荐其可能感兴趣的房源信息。

工业界除了将推荐系统应用到互联网之外，还通过竞赛的方式，推动工业界和学术界对推荐系统的研究和人才的培养，表 1-1 列出了近年来国内外有关推荐系统方面的竞赛情况。推荐系统大赛频繁举办，说明了工业界对推荐系统的重视。工业界举办各种推荐系统大赛有利于吸引更多地科研能力投入到推荐系统研究领域中来，有助于推荐系统人才的培养，从而更进一步地促进推动推荐系统的快速发展。

表 1-1 近年推荐系统大赛

时间	竞赛名称	最高奖金
2006	Netflix 推荐系统大赛	$100 万
2011	Yahoo 赞助的 KDD Music 推荐大赛	$5000
2012	Tecent 赞助的 KDD Weibo 推荐大赛	$5000

2012	Kaggel 职位推荐系统大赛	$10 万
2013	Baidu 电影推荐大赛	￥1 万

1.3.2 个性化推荐系统的研究

推荐系统除了具有较大商业价值而受到工业界的广泛关注和应用外，还因具有较大的理论价值而备受学术界的关注，产生了许多以推荐系统为主题的学术会议和期刊专栏，专门研究推荐系统。

1996 年 UC-Berkeley 学术界和商业界的研究者明确定义了协同过滤算法并探讨了其应用前景。

1997 年 Resnick[5]给出了电子商务个性化推荐系统的正式定义。该定义现在已被普遍采纳。

1998 年美国 AAAI 组织各国学者在 Wisconsin 举办推荐系统专门会议，就推荐系统未来发展方向进行了重点讨论[12]。

2001 年，纽约大学的 Adoavicius 和 Tuzhilin 实现了个性化电子商务网站的用户建模系统。

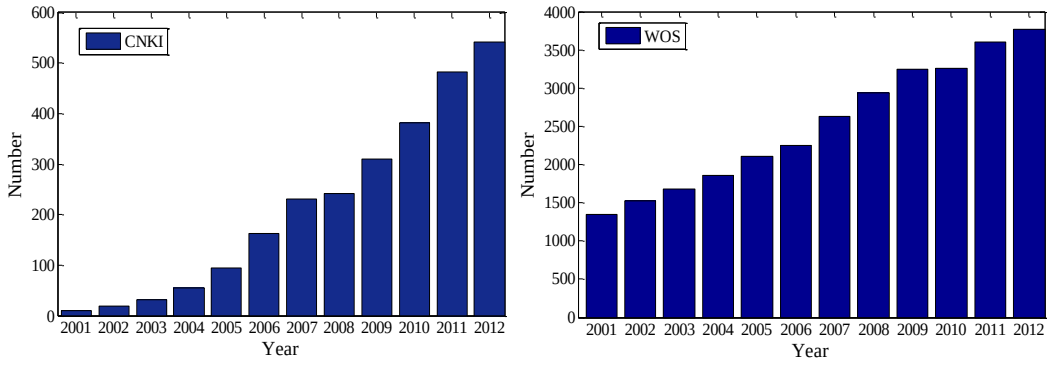
2006 年美国密西根大学开设了推荐系统课程。

2007 年 ACM 设立了推荐系统年会，至今每年均会举办一次。

2011 年我国工业界联合主办了第一届中国推荐系统大会。

2012 年举办了第二界中国推荐系统大会。

近年来，数据挖掘和机器学习领域，甚至物理领域的顶级期刊上推荐系统方面的文章也开始慢慢增多。伴随大量科学家投身到推荐研究领域，推荐系统的研究日趋完善，图 1-2(*a*)和图 1-2(*b*)分别是 Web of Science 和 CNKI 中，以“推荐系统”和“推荐算法”为关键词进行检索获得的在 2001-2012 期间发表的论文数量。从图中我们可以看出，早期国内论文数量较小，国外论文数据较多，说明国内个性化推荐研究起步要比国外晚。两图中显示的整体增长趋势表明近年来国内外个性化推荐方面的论文数量每年均有不同程度的增长，但是国外论文数量增长的速度没有国内快，说明近年来国内越来越多的学者开始关注个性化推荐，慢慢投入到个性化推荐系统的研究中来。



(*a*) (*b*)

图 1-2 CNKI 和 Web of Science 中推荐系统相关的论文数

除了大量学术论文，国内外很多学者整理出版了推荐系统方面的专著，见表 1-2。

表 1-2 国内外推荐系统方面的专著

时间	书名	作者
2009	集体智慧编程	Toby Segaran
2010	Mahout in Action	Sean Owen
2010	Recommender Systems: An Introduction	Jannach, Dietmar
2010	Recommender System Hand Book	Paul B. Kantor
2012	推荐系统实践	项亮
2013	推荐系统	詹尼士

推荐算法是推荐系统的核心，决定了推荐系统的推荐方式和推荐结果的准确性。根据推荐系统中推荐方式的不同，推荐系统通常分为如图 1-3 所示的几类。

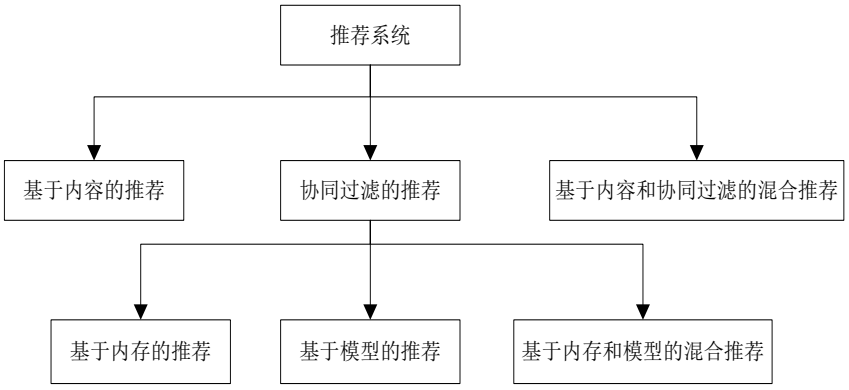


图 1-3 推荐系统的分类

1.4 本文研究的主要内容与创新点

本论文首先概况了个性化推荐系统在工业界的应用和学术界的研究背景及其国内外研究现状，总结了个性化推荐系统相关的理论与方法，综述了个性化推荐

系统中使用最为广泛的协同过滤算法，重点介绍了协同过滤算法的概念原理，然后综述了协同过滤算法的改进策略，在此基础上，从邻近选择和相似度计算两个角度提出了相应的改进算法。

（1）考虑负相关性信息的协同过滤算法。传统协同过滤算法采用 Pearson 相关系数计算相似度，在选取最近邻居时，仅考虑了正相关性信息，未考虑负相关性信息。为此本文提出了一种考虑负相关性信息的协同过滤算法，该算法选取正相关用户作为最近邻居，负相关用户作为最远邻居，使用参数调节最近邻居和最远邻居在推荐过程中的作用。我们在 MovieLens 数据集上的数值试验表明，负相关性信息有助于提高协同过滤算法的准确性和多样性。进一步分析发现，负相关性信息还可以大幅度提高度小用户的推荐准确性，故负相关性信息有助于解决冷启动问题和同时保证推荐结果准确性和多样性的问题。

（2）基于 Sigmoid 权重相似度的协同过滤算法。为了解决传统协同过滤算法在稀疏数据条件下相似度计算不准确，无法发现有效最近邻问题，提出了基于 Sigmoid 权重相似性的协同过滤算法。首先计算用户间的共同评分次数，然后使用经 Sigmoid 函数调整后的共同评分数加权相似度，产生更准确有效的最近邻。我们在 MovieLens 数据集上的实验表明，该算法不仅能获得比传统协同过滤算法更好的预测准确性和推荐覆盖率，而且能弥补权重相似度需要手动调节参数的不足。进一步分析发现，该算法还能提高度小用户的预测准确性。该工作表明 Sigmoid 权重相似度能有效缓解数据稀疏性问题和冷启动问题。

1.5 论文的组织结构

本文共分为六个部分，各章的主要内容如下：

第一章：绪论。首先对本文进行研究所处的背景进行了深入的介绍，然后提炼了个性化推荐在理论和实践两方面的意义，接着总结了推荐系统在国内外工业界的应用和学术界的研究状况，最后归纳了本文的主要研究内容和创新点。

第二章：个性化推荐算法理论与方法介绍。首先介绍了几种常见个性化推荐算法的相关理论和原理，包括关联规则推荐、内容推荐、协同过滤推荐、混合推荐和基于网络结构的推荐，然后总结了上述算法的优缺点。

第三章：协同过滤算法相关理论与方法。首先阐述了协同过滤算法的基本概念和原理，介绍了几种经典的协同过滤算法，如基于用户的协同过滤算法、基于项目的协同过滤算法和基于模型的协同过滤算法，最后从协同过滤算法的三个步骤入手，分别综述了相应的改进策略。

第四章：考虑负相关性信息的协同过滤算法。首先总结了国内外对协同过滤算法的研究现状，进而指出现有研究较少深入关注负相关性，然后在进一步分析

协同过滤算法不足的基础上，提出了考虑负相关性信息的协同过滤算法，给出了最远邻居的定义和新的评分预测公式，最后在 MovieLens 数据集上验证了负相关性信息既能提高预测的准确性，还能增加推荐列表的多样性，而且还能大幅度提高度小用户的准确性。

第五章：基于 Sigmoid 权重相似度的协同过滤算法。首先在综述协同过滤算法现有研究现状的基础上，指出传统相似度和现有权重相似度的不足，分析了 MovieLens 数据集上用户间的共同评分数分布及其与相似度之间的关系，在此基础上，提出了基于 Sigmoid 权重相似度的协同过滤算法，给出了新的权重相似度计算公式，并在 MovieLens 数据集上进行了验证，发现 Sigmoid 权重相似度在准确性和覆盖率上能取得较 Pearson 相似度和 Min 权重相似度更好的结果。

第六章：总结与展望。概述了本文提出的两个算法的主要过程和结论，指出了可供继续研究的几个方向，同时展望了协同过滤算法的几个改进方向，供以后进一步的研究。

本文的组织结构如图 1-4 示

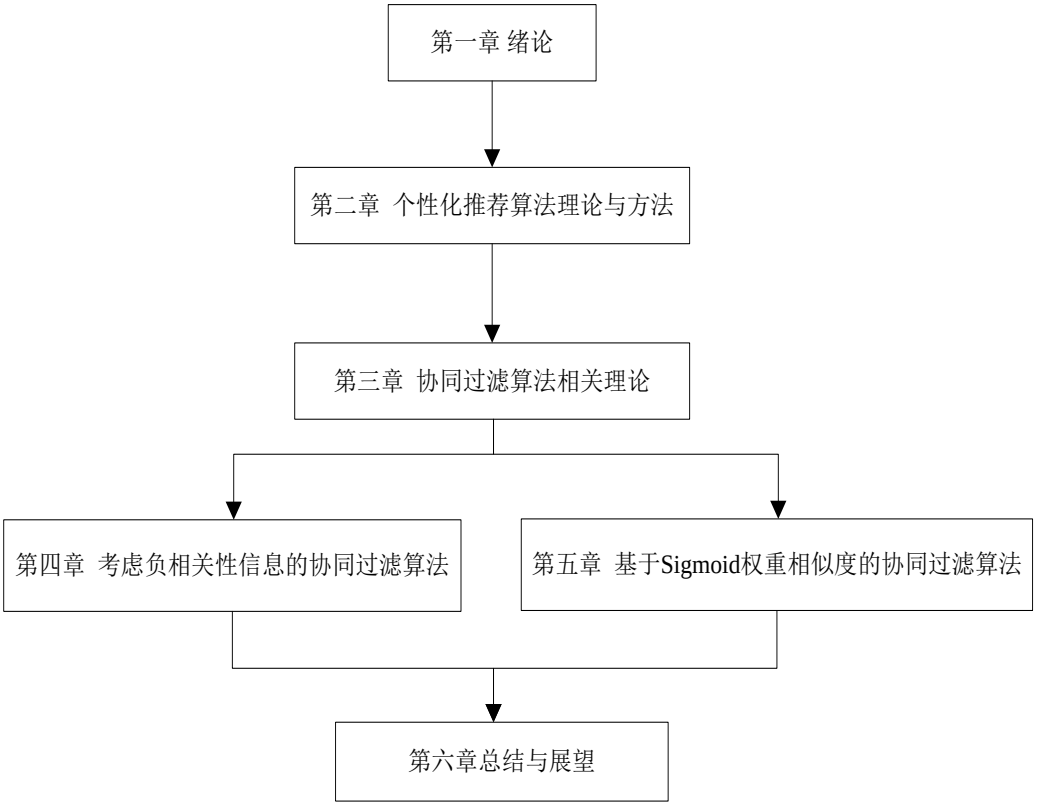


图 1-4 本文组织结构图

第二章 个性化推荐理论与方法

个性化推荐技术能够通过分析用户行为数据，学习用户兴趣，实现个性化的精准推荐。个性化推荐技术有助于互联网公司提高交叉销售和用户点击转化率，增加用户与网站的交互。个性化推荐系统已经成为互联网公司贯穿业务的一个标准配件，不在可有可无，而是必不可少。本章将介绍个性化推荐相关的理论与方法。

2.1 基于关联规则的推荐

关联规则是一种重要的数据挖掘方法，常用于发现数据集中项目元素间的关联关系，在电子商务网站中广泛用于购物篮分析和挖掘用户购买行为中的关联关系。

关联规则由规则体和规则头组成，一般将已经购买的商品作为规则头，打算推荐的商品作为规则体，构成由规则头推荐规则体的结构。一条关联规则可以很形象的描述成购买了该商品的用户同时也购买了另一种商品。著名的啤酒和尿布的故事就是关联规则最典型最成功的例子。如果商品 A（如啤酒）在商品 B（如尿布）购买的情况下也购买了很多，我们就将它表达成一条关联规则 $A \Rightarrow B$，其中 A 为规则头，B 为规则体。

通常采用如下方式描述关联规则：

设项集 $I=\{i_1,i_2,...,i_N\}$，D 为事务数据库，事务 T 为不同项的集合，其中 $T \subseteq I$。记项集为 A，其中如果满足 $A \subseteq T$，则说明事务 T 包含 A，那么就有关联规则 $A \Rightarrow B$，其中 $A,B \subseteq I$ 并且 $A \cap B = \varnothing$。

关联规则 $A \Rightarrow B$ 在数据集 D 上的支持度为 D 中事务包含 $A \cup B$ 的比例：

$$support(A \Rightarrow B) = P(A \cup B) = \frac{count(A \cup B)}{|D|} \tag{2-1}$$

关联规则 $A \Rightarrow B$ 在数据集 D 的置信度为 A 中事务同时包含事务 A 和事务 B 的比例：

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{count(A \cup B)}{count(A)} \tag{2-2}$$

如果 $support(X)$大于等于最小支持度，则称 X 为频繁项目集，否则为非频繁项目集。

支持度用于度量关联规则在事务数据库中的统计重要性，说明关联规则在已

发生事务中的代表性，而置信度用于衡量关联规则的准确性。满足最小支持度和置信度的关联规则称为强关联规则。若 *support*(啤酒⇒尿布)=20%，*confidence*(啤酒⇒尿布)=80%，表明在数据库中 20%的事务同时包含了啤酒和尿布，而购买啤酒的顾客中有 80%的人会去购买尿布。事务数据库中挖掘关联规则，通常采用如下步骤：

1、 产生所有的频繁项集。频繁项集即为所有满足最小支持度阈值的项集。

2、 由频繁项集导出强关联规则。在频繁项集中找出所有满足最小置信度阈值的项集。

关联规则算法最早是由 Agrawal[13]等人提出，其中使用的最多的算法是 Apriori 算法[14]，后来许多专家提出了各种改进的算法，如 Hash 表法[15]，FP-Growth[15]等。

关联规则推荐是指使用关联规则挖掘算法分析数据库中当前用户的历史交易记录，挖掘用户已购买产品间的关联关系，发现有价值的商品关联组合，即超过最小支持度和最小置信度阈值的商品关联组合，进而将关联组合中用户还未购买的商品推荐给用户，实现个性化推荐的功能。关联规则用于个性化推荐，不仅可以为用户提供其感兴趣的商品，降低用户搜索商品的成本，还有助于电子商务平台发现商品间的相关关系，从而优化网站布局。

Fu、Budizk 和 Hammond 等最早将关联规则用于推荐系统[16,17]，利用 Apriori 算法挖掘用户浏览行为中的关联规则，从而产生推荐。关联规则可以提前计算好，故通常在服务器端离线进行，以保障推荐系统的实时性[18]。

关联规则推荐的核心是关联规则的挖掘，推荐效果依赖于关联规则的数量和质量，而支持度的大小对关联规则生成有较大影响，设置得过大或者过小，将会产生过少或过多的关联规则，而且得到很多与目标用户无关，不能用于推荐的关联规则，严重降低了推荐系统的质量。Lin[19]等人提出改进的关联规则挖掘算法，无需预先设定最小支持度，而是限制返回的关联规则的数目，自动调节最小支持度，保证规则数量在设定的范围内。

关联规则推荐的最大缺点就是可扩展性问题，即随着推荐系统用户数和项目数的飞快增长，关联规则的数量增多，系统将变得难以管理[20]。除了关联规则的数量呈爆炸式增长外，大量关联规则中还会包含冗余关联规则。比如规则“购买面包的顾客还会同时购买牛奶和黄油”，可以导出四个相关规则：顾客购买面包后还会购买牛奶、顾客购买面包后还会购买黄油、顾客购买面包和牛奶后还会购买黄油、顾客购买面包和黄油后还会购买牛奶。我们只需第一条规则即可，其余四条规则都是冗余关联规则。冗余关联规则对推荐系统毫无益处，既增加了关联规则

挖掘的时间，又给推荐系统带来了不必要的负担，降低推荐的效果。

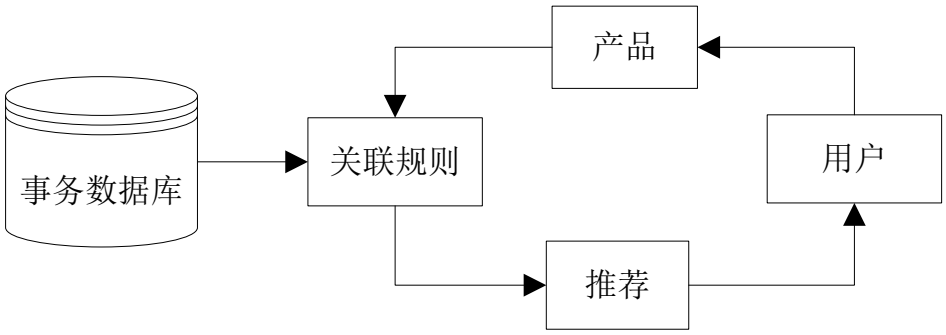


图 2-1 关联规则推荐示意图

2.2 基于内容的推荐

基于内容的推荐(content-based recommendation)[21,22]是早期推荐系统中运用比较成功的一种方法，该方法首先分析用户已选择的项目，建立用户兴趣偏好，同时抽取和表示项目内容特征,根据用户兴趣和项目特征的相似性对用户进行推荐，将相似度高的项目推荐给当前用户。比如当用户在网上购买了一本推荐系统方面的书，那么基于内容的推荐系统会给他推荐另一本关于推荐系统方面的书。

基于内容的推荐通过计算项目的内容特征和用户兴趣偏好的相似性进行推荐，一般分为用户兴趣和项目特征内容的建立，用户兴趣和项目内容相似性的计算，用户兴趣相似项目的推荐三个步骤。

用户兴趣模型的建立：用户兴趣模型通常采用基于行为的模型和基于兴趣的模型[23,24]。反映用户兴趣的数据可以通过显性和隐性两种方式获得。显性方式比如当用户注册成为电子商务网站会员时，往往会提供人口统计学方面的信息，或者电子商务网站通过发布调查问卷，收集用户反馈。隐性方式比如通过文本挖掘等技术分析用户的访问日志，Web 挖掘等分析用户浏览记录等方式获得。隐性方式获得的结果通常比较多但是准确性不够好，显性方式获得的结果通常比较少但是准确性较高。

一般我们可以通过向量表示用户的兴趣特征，即兴趣特征向量。用户 u 的兴趣特征向量表示为 Profile(u)={w_{u1},w_{u2},$\cdots$,w_{uk}},w_{uk} 表示用户 u 对关键词集合中第 k 个词的喜欢程度。

项目内容特征的建立：音频视频等资源不易提取内容特征，因此基于内容的推荐主要用于文档资源。在互联网下，文档内容信息较多，内容差别也较大，研究中通常引入自然语言处理技术从文档中抽取关键词，进行文档特征化，然后使用向量空间模型形成关键词向量来描述文档。文档经过特征提取后，需要计算每个特征的权重，通常采用信息检索领域最经典的词频-倒排文档频率(term

frequency-inverse document frequency，简称 *TF-IDF*)[25]。词频 *TF*(Term Frequency)表示该特征在文档中出现的次数，*IDF*(Inverse Document Frequency)表示 *log*(所有文档数/包含该特征的文档数)。*TF-IDF* 的定义如下：设有 *N* 个文本文档，特征词 k_i 在 *n* 个文档中出现，f_{ij} 为特征词 k_i 在文档 d_j 中出现的次数，那么特征词 k_i 在文档 d_j 中的词频为：

$$TF_{ij} = \frac{f_{ij}}{\max(f_{zj})} \tag{2-3}$$

其中 f_{zj} 为任意特征词 k_z 在文档 *j* 中出现的次数。

IDF_i 定义为：

$$IDF_i = \log\frac{N}{n} \tag{2-4}$$

因此文档 d_j 可以表示为向量模型 $d_j=(W_{1j},W_{2j},\ldots,W_{3j})$，其中 W_{ij} 定义为

$$W_{ij} = \frac{f_{ij}}{\max f_{zj}}\log\frac{N}{n} \tag{2-5}$$

相似性计算：建立用户兴趣模型和项目的特征描述后，我们可以计算用户兴趣特征向量和项目特征向量间的相似性，以此表示用户对该项目的感兴趣程度。按照该思路，计算给定用户与所有未曾看过的项目资源间的相似性，从中选取相似度最高的 N 个项目返回给特定用户。相似性计算一般采用余弦相似度公式：

$$Similarity(i,j) = \frac{Vec_i \times Vec_j}{\|Vec_i\| \times \|Vec_j\|} \tag{2-6}$$

其中 Vec_i 表示项目 *i*，Vec_i 表示项目 *j*。

基于内容的推荐具有如下优点：

(1) 简单，高效。

只需计算用户兴趣特征向量和项目特征向量的相似性即可做出推荐。

(2) 能发现隐藏的暗信息。

电子商务网站中，有很多产品是被较少用户选择过，我们将这些产品称为非流行的小众商品，基于内容的推荐通过利用项目固有的内容信息，能够将这些隐藏在系统中不易被用户轻易发现的暗信息推荐出来，展示给用户。

(3) 能提供推荐解释，增加推荐系统的透明性。

基于内容的推荐系统能够对推荐结果给出解释，指明推荐是根据项目的哪几个特征属性计算而来的，可以增加推荐系统的透明性和用户对推荐系统的信任。

基于内容的推荐有如下缺点：

(1) 推荐对象有限，无法推荐非结构化的对象。

基于内容推荐的核心在于关键词的提取，因此容易受特征提取技术的限制。不能推荐一些无法进行特征提取的对象，较大的影响了推荐系统的使用范围。

(2) 未考虑用户行为信息。

基于内容的推荐仅考虑用户和项目本身的特征，未考虑用户行为，而用户行为信息能够较好地反映用户的兴趣偏向。

(3) 过度专一化。

基于内容的推荐会经常给用户推荐一些过去喜好的物品相似的物品，较难推荐用户兴趣范围以外的商品，大大缩小了用户的视野，比如新闻推荐中容易推荐同一事件的不同新闻。在电子商务领域中，这一点也非常不利于挖掘客户的潜在需求。

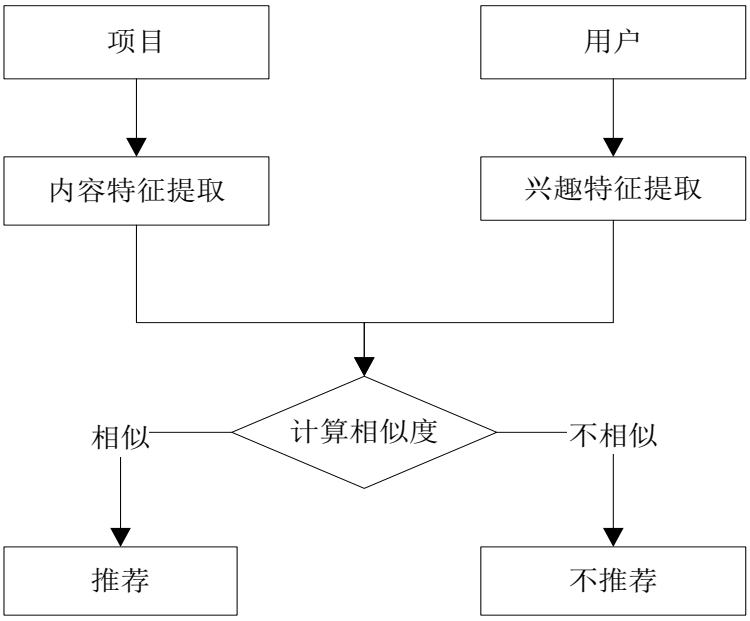


图 2-2 基于内容的推荐算法示意图

2.3 协同过滤推荐技术

协同过滤(Collaborative Filtering)也称为是协作过滤，社会过滤，是学术界研究最广泛，工业界使用最流行的个性化推荐算法。该算法借鉴群体智能的思想[26]，模拟人的从众心理，根据“物以类聚，人以群分”，将与当前用户兴趣相同的其他用户喜欢的商品推荐给当前用户，比如现实生活中的朋友或者兴趣相近者的意见对我们有较大的说服力，我们通常会认真考虑这部分人所给出的推荐。

协同过滤算法最早是在 1992 年由 Goldberg 等人[2]首先提出，并应用在邮件推荐系统 Tapestry 中，以解决研究中心资讯过载的问题。此后另一个里程碑就是

GroupLens 研究小组在 1994 年提出了基于用户的协同过滤算法，受到学术界的广泛关注。该算法用于 GroupLens 新闻推荐系统中，随后出现了很多与之相似的推荐系统，比如 MovieLens 电影推荐系统、Ringgo 音乐推荐系统和 Jester 笑话推荐系统等。GroupLens 研究小组免费公开了 MovieLens 数据集，该数据集被学术界和工业界广泛用作算法测试和改进，极大地促进了推荐系统领域的大繁荣。随后众多互联网公司或科研单位也纷纷效仿，公开了自己的数据集，比如 Netflix，BookCrosing 等。随后 Amazon 在 2003 年提出的基于项目的协同过滤算法，不仅奠定了 Amazon 推荐之王的地位，而且还为其贡献了 35%的利润。1994 年到 2006 年之前，推荐算法的研究主要集中在基于内存的协同过滤算法，包括基于用户的协同过滤算法和基于项目的协同过滤算法。2006 年 Netflix 竞赛，悬赏百万美元，促进了学术界和工业界更进一步地提高协同过滤算法的精度,产生了很多新算法，其中基于矩阵分解(SVD) 模型的协同过滤算法得到了学术界的广泛关注。

协同过滤算法根据推荐系统中所有用户的历史信息，计算目标用户和其他用户的兴趣相似性，选取与目标用户兴趣相似度较高的若干用户作为目标用户的最近邻居，由于目标用户的最近邻居与其有相近的兴趣偏好，因此推荐系统会将最近邻居喜欢的物品推荐给目标用户。不受推荐对象的限制，既能推荐文本内容等结构化对象，又能推荐音频、视频等非结构对象，是协同过滤算法最大的优点，因此成为工业界应用最成功，学术界研究最广泛的个性化推荐算法。表 2-1 列出了国内外使用协同过滤算法的网站。

表 2-1　国内外使用协同过滤算法的网站列表

推荐系统名称	推荐产品	网站
CD Now.com	CD	http://www.cdnow.com
Movielens	电影	http://moivelens.umn.edu
Grouplens	Usernet 新闻	http://www.grouplens.com
Amazon	图书	http://www.amazon.com
Reel	电影	http://www.reel.com
Jester	笑话幽默	http://shadow.ieor.berkeley.edu/humor
Internet Watcher	网页	http://www.internetwatcher.com
Phoaks	网站	http://www.poaks.com
Yenta	寻友	http://foner.www.media.mit.edu
豆瓣	音乐图书	http://www.douban.com/
安居客	房地产	http://shanghai.anjuke.com/

协同过滤算法具有如下优点：

(1) 协同过滤算法可以应用在计算机不易处理的复杂非结构化对象上，其原因是协同过滤算法不计算项目的内容属性，而是根据用户行为中的群体智慧来进行推荐。

(2) 协同过滤算法能够推荐新项目，发现隐藏在海量信息中的“暗信息”和海量商品中的长尾物品。基于内容的推荐将与目标用户已购买或选择的商品相似的商品推荐给目标用户，产生的推荐结果通常是目标用户比较熟悉的项目。协同过滤算法基于用户行为计算用户与用户，项目与项目间的相似性，所以能够推荐内容上完全不相关的项目，充分挖掘用户的潜在兴趣，给用户带去惊喜性。

(3) 协同过滤算法能有效减少用户的反馈量。协同过滤算法从用户的行为中抽取用户兴趣，用户行为包括搜索浏览行为，购买历史或商品评分信息，整个过程完全不会干扰用户的访问。

协同过滤算法作为一种广泛使用的个性化推荐算法，仍然存在很多问题和挑战，其中比较典型的有稀疏性问题、冷启动问题、扩展性问题和实时性问题等。

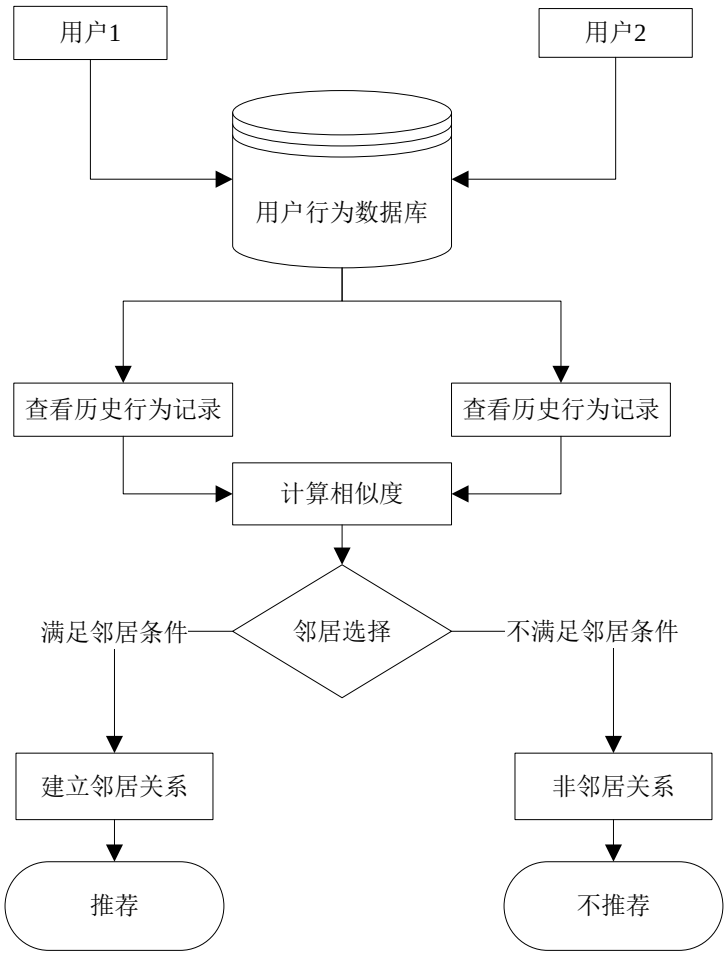


图 2-3 协同过滤算法流程示意图

2.4 混合推荐技术

上述介绍的几种推荐算法，无论是基于内容的推荐还是关联规则推荐，抑或是协同过滤推荐，都存在这样或那样的缺陷，一种推荐技术在某种情况下取得不错的效果，但不能保证在所有评价指标上均取得领先优势，于是很多科研人员提出将多种推荐算法进行组合，从而优势互补，这就是混合推荐技术的基本思想。协同过滤推荐和内容推荐的组合[21,27~29]是企业应用和学术研究过程中使用最多的两种混合策略。

主要的混合策略有[30]：

(1) 混合集成：这是一张最简单的混合方法，分别用多种推荐算法产生结果，然后利用某种算法把推荐结果混合在一起作为最终的推荐结果。

(2) 加权集成：给不同的推荐技术赋予不同的权重，将不同推荐技术预测的评分加权和作为最终的预测评分，权重的设置会根据不同的条件。

(3) 转换集成：根据环境的不同选择不同的推荐技术，比如推荐新闻时根据内容推荐，推荐电影时用协同过滤推荐。

(4) 瀑布集成：用一种推荐技术优化另一种推荐技术的推荐结果，过滤掉某种推荐技术中不太符合要求的推荐结果。

(5) 特征增值集成：一种推荐技术的推荐结果作为另一种推荐技术的输入。

除了协同过滤算法和基于内容的混合推荐算法外，最近兴起的基于网络结构的推荐算法中，也存在一类混合算法[31]，该混合的方式是将基于热传导的推荐算法和基于物质扩散的推荐算法相混合，基于热传导的推荐算法专门用来推荐不太流行的冷门资源，而基于物质扩散的推荐算法的准确性较高，两者结合以后，能同时提高推荐的准确性、新颖性和多样性。

混合推荐技术的初衷是优势互补，但是在实际应用中难以寻找到一种较好的集成策略。同时由于推荐系统需要多个推荐技术参与计算，无疑会增加计算复杂性，在一定程度上会降低推荐系统的效率和实时性[32]。

2.5 基于网络结构的推荐

用户和产品是推荐系统中的主要元素，任何推荐系统的主要目的就是寻找用户和产品间的关系，为用户寻找其可能喜欢的产品或者是将产品推荐给可能喜欢的用户。现有的推荐系统大都是利用用户对产品的行为，比如打分、评论、关注、收藏和下载，来表示用户兴趣模型，建立用户-产品间的关系，而用户的这些行为信息还可以很容易的表示成图或网络，这就是基于网络结构的协同过滤算法(network-based collaborative filtering)。NBCF 算法将用户和产品表示为点，将用户和产品之间的关系表示为边。由于推荐系统仅涉及用户和产品二元对象，因此可

以通过用户对产品的行为建立用户-产品二部图来表示推荐系统。

基于网络结构的推荐算法是由周涛[33,34]、Huang 等人[35]提出的，后来周涛在二部图上提出了全新的资源分配算法。张翼成[36,37]考虑用户的打分信息，将物理学中的物质扩散和热传导运用到二部图中，提出了基于物质扩散和热传导的个性化推荐算法。Liu 等人[38]通过考虑用户产品间的二阶关联信息，发现通过降低主流偏好可以提高推荐算法的准确性。Liu 等人[39]还提出了基于用户兴趣点的物质扩散算法，明显提高了推荐算法的准确性和推荐列表的多样性。下面介绍基于物质扩散的个性化推荐算法。

设用户集合为 $U = \{u_1,u_2,\ldots,u_n\}$，产品集合为 $O = \{o_1,o_2,\ldots,o_m\}$，那么推荐系统可以表示为邻接矩阵 $A = \{ a_{ij}\} \in R^{m,n}$，其中如果用户 i 选择过产品 j 那么 $a_{ij}=1$，否则 $a_{ij} = 0$。物质扩散推荐算法的基本思想是目标用户选择过的产品都具有某种向该用户推荐其他产品的能力。算法的原理图见图 2-4。

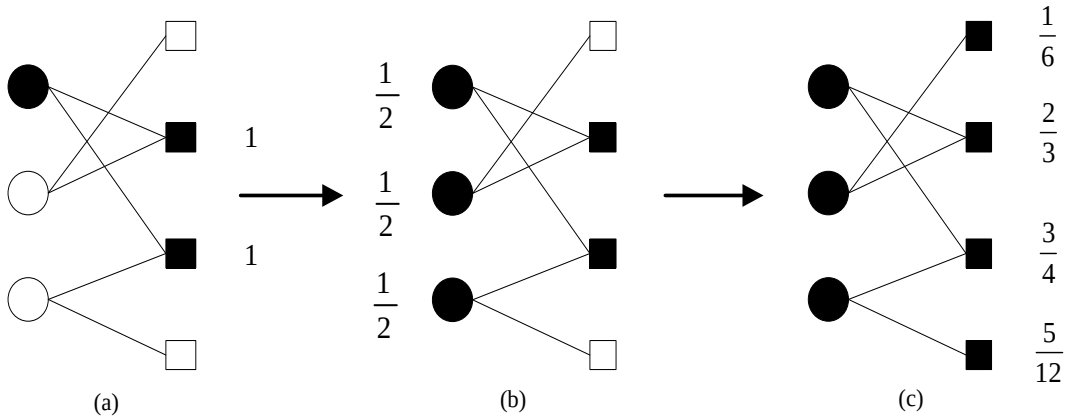


图 2-4 基于物质扩散的推荐算法

图中圆圈表示用户，方块表示产品，连边表示该用户选择过该产品。假设用户 j 选择过的产品都具有向该用户推荐其他产品的能力，并赋予 1 单位的初始资源，物质扩散通过三步扩散来实现：

第一步：目标用户 j 选择过的产品被激活并赋予 1 单位的初始资源，见图 2-4(a)，

第二步：目标用户 j 选择过的产品按照资源评价分别的原则，将自己用户的资源平均分配给选择过该产品的用户，见图 2-4(b)，

第三步：所有用户所拥有的资源再评分分配给被这些用户选择过的所有的产品，具体步骤见图 2-4(c)。对于任意的用户 j，经过三步扩散以后，系统对用户 j 没有选择过的产品按照资源的多少由高到低排列，将排名靠前的产品推荐给用户 j。

由上述三步扩散的过程可知，产品 j 分配给产品 i 的资源为

$$w_{ij} = \frac{1}{k_j} \sum_{l=1}^{n} \frac{a_{il} a_{jl}}{k_l} \tag{2-7}$$

其中 k_j 表示产品 j 的度，k_l 表示用户 l 的度，产品相似度矩阵可表示为

$$W = \{w_{ij}\}_{m \times n} \tag{2-8}$$

对于一个给定的目标用户 j，初始资源可以代表他的个性化信息，可以用一个 m 维的 0/1 矢量来表示 $f=\{f_1,f_2,\ldots,f_m\}^T$，其中 $f_i = a_{ij}$。最终的资源分配矢量为

$$f = Wf \tag{2-9}$$

将对应的产品根据其获得资源的多少，由高到低地降序排序，从中选取目标用户没有选择过的排名靠前的产品推荐给用户。

无论是关联规则推荐，传统的基于内容的推荐，经典的协同过滤推荐抑或是最近兴起的基于网络结构的推荐等，都有各自的优点和缺点，详细总结如表 2-2 所示。

表 2-2 各种推荐算法的优缺点比较

推荐算法	优点	缺点
基于关联规则的推荐	对项目内容的依赖性较小	初期规则不易获取，后期不易管理，出现规则爆炸，易出现冗余规则，规则更新较慢，不能算是个性化推荐
基于内容的推荐	简单，高效，推荐结果直观，能够发现隐藏的暗信息，能提供推荐解释，增加推荐系统的透明性，不受稀疏性和项目冷启动问题的影响，对文本资源推荐效率高	受特征提取技术的困扰，推荐对象有限，不适合推荐多媒体资源，无法对非结构化的物品产生推荐，未考虑用户行为信息，过度专一化，不易发现用户潜在的兴趣爱好

协同过滤推荐	可以应用在计算机不易处理的复杂非结构化对象上，能够推荐新项目，发现隐藏在海量信息中的“暗信息”和海量商品中的长尾物品，协同过滤算法能有效减少用户的反馈量	存在稀疏性和冷启动问题，推荐结果不易解释，推荐效果不稳定，容易受数据集的不同影响
混合推荐	吸取基于内容的推荐和协同过滤算法两者均有的优点	实现复杂度高，计算量大，系统的实时性会受到影响
基于网络结构的推荐	能够推荐新信息，帮助用户发现潜在的但是用户未曾发现的物品；应用广泛，不受物品的内容限制；计算复杂性低	受冷启动问题的困扰，无法为新用户推荐产品，无法将新产品推荐给用户；数据稀疏性对算法精度有一定的负面影响

在实际的推荐系统开发过程中，应该根据各自的业务特点，选择适合自身业务特性的个性化推荐算法，或者建立个性化推荐算法库，结合多种个性化推荐算法。

2.6 本章小结

本章介绍了几种常见的个性化推荐算法，描述了关联规则推荐算法、基于内容推荐算法、协同过滤推荐算法、混合推荐算法以及基于网络结构的推荐算法的基本思想和原理，给出了关联规则推荐、内容推荐和协同过滤推荐算法的示意图，最后总结了上述几种个性化推荐算法的优缺点。

第三章 协同过滤算法相关理论

3.1 协同过滤算法的概念和原理

协同过滤(Collaborative Filtering)最早是有 Goldberg[2]提出的，他们利用系统中的其他用户来为当前用户过滤信息。协同过滤算法一经提出，便受到了学术界的重点关注和工业界的广泛应用，每年 ACM 主办的推荐系统大会上，均有大量来自学术界和工业界的论文和研讨会，许多著名的互联网公司比如 Amazon、eBay 等都是使用协同过滤算法的典型代表。

协同过滤算法的基本思想是根据当前用户的行为，寻找与之兴趣相似的用户，然后将这些用户喜欢的并且当前用户又没有接触过的商品推荐给当前用户。其基本思想非常直观，比如今天你想看电影但是又不知道哪部电影好看，这个时候你可以去找一部与以前看过电影类似的电影，比如都是喜剧片，或者都是某个演员主演的，此外你还可以去找自己的亲朋好友，问问他们最近在看什么电影，让他们给你推荐推荐。第一种情况属于基于内容的推荐，第二种情况便是基于协同过滤的推荐。协同过滤算法丝毫不用考虑任何的项目内容信息，能够推荐音频视频图片等复杂非结构化的对象，而且完全根据最近邻居的偏好产生推荐，从而能够将目标用户不熟悉的物品推荐给用户，帮助用户发现新的兴趣点，让推荐系统比用户自己还要更懂自己，因此我们可以发现协同过滤算法的基础假设为

(1) 用户的兴趣是复杂多样的，但是某些用户间的兴趣具有相似性；

(2) 用户对不同项目的评价行为能够反映用户的兴趣偏好；

(3) 用户会对未知项目给出与其兴趣相似用户一致的评价。

3.2 经典的协同过滤技术

Breese 等人将协同过滤算法分成了两类，一类是基于记忆的（最近邻）协同过滤算法，另一类是基于模型的协同过滤算法[40]。基于记忆的协同过滤算法是推荐系统中最基本的算法，得到了学术界的深入研究和工业界的广泛关注，该算法直接使用系统中存储的用户项目评分计算相似度，然后使用 KNN 算法寻找最近邻，并以此预测目标用户未评分项目的评分。根据相似度计算的角度不同，基于近邻的协同过滤算法分为基于用户的协同过滤算法和基于项目的协同过滤算法。基于模型的推荐算法不是直接根据用户项目评分进行推荐而是根据统计技术或者机器学习算法对用户项目评分矩阵进行训练所得到的模型进行推荐。协同过滤算法的分类见图 3-1 所示。

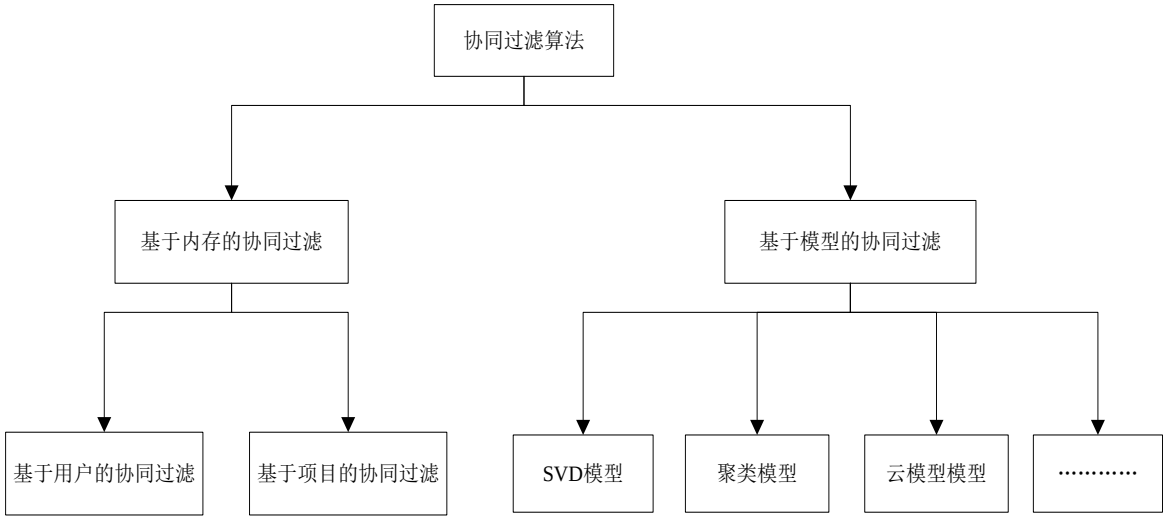


图 3-1 协同过滤算法分类

3.2.1 基于用户的协同过滤算法

基于用户的协同过滤算法是推荐系统中最早的个性化推荐算法，个性化推荐系统的诞生就是以该算法的诞生为标志。如果用户对某些项目的评分比较相似，则说明这些用户的兴趣爱好相近，那么他们对其他项目的评分也会比较相近，因此基于用户的协同过滤算法首先计算用户间的兴趣相似性，据此找到与当前用户兴趣最相似的若干用户作为最近邻居，然后根据最近邻居预测其未评分项目的评分，最后选择预测评分最高的项目组成推荐列表。该过程如下图所示。

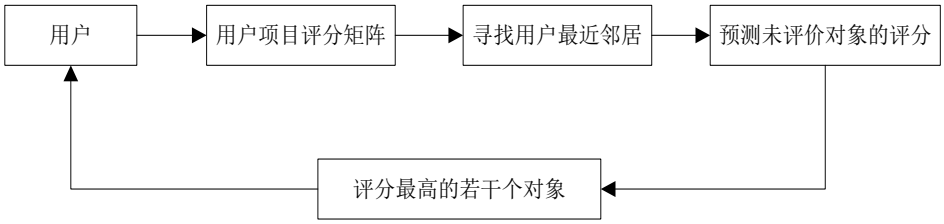


图 3-2 基于用户的协同过滤算法示意图

基于用户的协同过滤算法主要有如下几个步骤：

(1) 数据初始化

数据初始化就是用户项目评分矩阵的初始化。用户对项目的评分包括显式评分和隐式评分，显式评分即为用户对项目的直接打分，隐式评分即为根据用户的历史行为记录，如浏览点击时间和频率等方式推断出的评分，可见显式评分量少而质量更可靠，隐式打分量大但质量不可靠，表 3-1 是显式数据和隐式数据的比较。

表 3-1 显性反馈数据和隐性反馈数据的比较

	显性反馈数据	隐性反馈数据
用户兴趣	明确	不明确
数量	较少	较多
存储	数据库	分布式存储系统
实时读取	实时	有延迟
正负反馈	都有	只有正反馈

评分值通常有不同的形式，比如用 0 和 1 表示用户对项目喜欢/不喜欢，购买/不购买等，此外还有用具体的评分等级数据来表示用户的的喜好程度，比如 MovieLens 使用 5 分制，0 分表示用户没有对电影给出评分，1 到 5 表示用户对电影的喜欢程度，分数越高表示用户越喜欢。有时为了统一，可以将评分制转化为二进制，比如将大于评分范围中位数的评分取为用户喜欢，小于中位数的取为不喜欢。根据结合显式评分和隐式评分可以得到一个 N×M 的用户项目评分矩阵 R，如式(3-1)所示。

表 3-2 用户项目评分矩阵

	$Item_1$	...	$Item_j$	...	$Item_n$
$User_1$	R_{11}	...	...	...	...
...	...	...	...	...	...
$User_i$	R_{i1}	...	R_{ij}	...	R_{in}
...	...	...		...	...
$User_m$	R_{m1}	...	R_{mj}	...	R_{mn}

其中 M 代表用户数，N 代表项目数，行向量表示用户对项目的评分，纵向量表示对该项目进行过评分的用户。若用户 i 对项目 j 有过评分则 R_{ij} 不等于 0，否则 $R_{ij}=0$。

(2) 最近邻居形成

基于用户的协同过滤算法的核心是利用最近邻居做出推荐，因此邻居形成是协同过滤算法最关键的步骤。对于目标用户 u，协同过滤算法寻找与其相似度最高的 K 个用户作为 u 的最近邻居集合，$N(u)=\{u_1,u_2,...,u_k\}$，$u\in N(u)$，其中最近邻居集合中的用户按照相似度 $Sim(u,u_1)$由大到小排序。$Sim(u,u_k)(1<k<K)$取值范围为[-1,1], $Sim(u,u_k)$取值越接近 1，说明 u 和 u_k 的兴趣越相似，$Sim(u,u_k)$取值越接近-1，说明 u 和 u_k 的兴趣差异越大。

用户间的相似度是根据用户项目评分矩阵计算的。常用的计算方法有余弦相

似性及其修正相似性，Pearson 相关相似性[41,42]。

a) 余弦相似性

余弦相似性用 n 维向量对用户评分进行建模，如果用户没有对项目评分，那么用户对该项目的评分置为 0。用户间的相似性用 n 维用户评分向量间夹角的余弦表示，夹角越小，用户间的相似度就越高。

设 I_u 和 I_v 为用户 u 和用户 v 各自的评分项目集合，I_{uv} 为用户 u 和 v 的共同评分项目集合，即 $I_{uv}=I_u \bigcap I_v$，向量 $\vec{u}$ 和 $\vec{v}$ 分别为用户 u 和 v 在 I_{uv} 上的评分，用户 u 和 v 在 n 维项目空间上的评分记为 R_{ui}，R_{vi}，则用户 u 和 v 之间的相似度 $sim(u,v)$ 为：

$$sim(u,v)=\cos(\vec{u},\vec{v})=\frac{\vec{u}\cdot\vec{v}}{\left\|\vec{u}\right\|\times\left\|\vec{v}\right\|}=\frac{\sum_{i\in I_{uv}}R_{u,i}R_{v,i}}{\sqrt{\sum_{i\in I_u}R_{u,i}^{\ 2}}\sqrt{\sum_{i\in I_v}R_{v,i}^{\ 2}}} \tag{3-1}$$

b) 修正余弦相似性

余弦相似性未考虑不同用户打分习惯的影响，比如有的用户比较仁慈，喜欢给高分，有的用户比较严格，即使很喜欢的项目所给的评分也不高，为此，我们在余弦相似性的基础上，在分子部分减去用户对项目的平均评分，就可以得到修正的余弦相似性，记用户 u 和用户 v 的平均评分分别为 $\overline{R_u}$ 和 $\overline{R_v}$，则用户 u 和 v 的修正余弦相似性为

$$sim(u,v)=\frac{\sum_{i\in I_{uv}}(R_{u,i}-\overline{R_u})(R_{v,i}-\overline{R_v})}{\sqrt{\sum_{i\in I_u}(R_{u,i}-\overline{R_u})^2}\sqrt{\sum_{i\in I_v}(R_{v,i}-\overline{R_v})^2}} \tag{3-2}$$

c) Pearson 相关相似性

Pearson 相关相似性用 Pearson 相关系数来表示用户间的相似性，相关系统越大，表示相关性越强，用户间的相似度就越高。与余弦相似性不同的是 Pearson 相关相似性是在用户间共同评分的基础上计算相似度。$\overline{R_u}$ 和 $\overline{R_v}$ 在此处表示用户 u 和用户 v 在两者共同评分项目集合 I_{uv} 上的平均评分。

$$sim(u,v)=\frac{\sum_{i\in I_{uv}}(R_{u,i}-\overline{R_u})(R_{v,i}-\overline{R_v})}{\sqrt{\sum_{i\in I_{uv}}(R_{u,i}-\overline{R_u})^2}\sqrt{\sum_{i\in I_{uv}}(R_{v,i}-\overline{R_v})^2}} \tag{3-3}$$

表 3-3 相似度计算方法比较

相似度计算方法	优点	缺点
Pearson 相似度	在两用户间的共同评分项目集上计算相似度，计算结果较准确	容易受数据稀疏性的影响，极端稀疏下，甚至无法计算相似度
余弦相似度	计算方法简单	没有考虑用户评分的统计

		特征
修正余弦相似度	在余弦相似度基础上考虑了减去用户的评均评分，调整了用户评分尺度的不一致性	不能很好地度量用户或项目间的相似性

使用上述相似度计算方法，可以计算系统中所有用户间的两两相似度，形成 m×m 用户相似度矩阵 $S(u,v)$，该矩阵是一个对称矩阵，对角线上的元素表示用户与用户自身的相似性，一般都令其等于 0，矩阵中的其他元素 $S(u,v)$为用户 u 和用户 v 之间的相似性。在用户相似度矩阵基础上，我们通常采用 K 近邻策略和阈值策略选取用户的最近邻居。

a) K 近邻策略

K 近邻(K-nearest neighbor)即 KNN，根据预先设定的邻居个数 K，从相似度矩阵中选取与当前用户相似度最高的前 K 个用户。

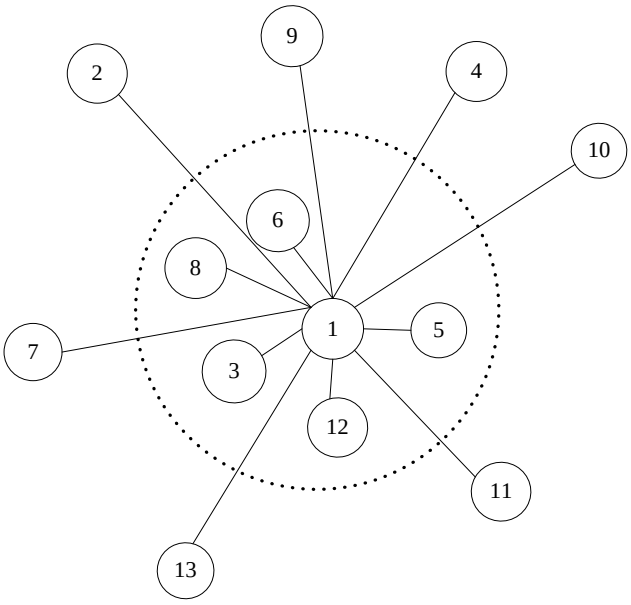


图 3-3 KNN 寻找邻居的示意图

b) 阈值策略

该方法预先设定相似度阈值，找出与目标用户相似度大于该阈值的所有其他用户。

c) 混合策略

该方法是前面两种方法的混合，首先采用阈值策略选取邻居集合 N，然后在从该集合中选取与目标用户相似度最高的 K 个用户，如果 N 中的用户数小于 K，那么就只选取这部分大于阈值的用户。

采用某种邻居选取策略，从用户相似度矩阵中选取目标用户的最近邻居集合

$N(u)=\{u_1,u_2,\ldots,u_k\}$。

(3) 预测评分及产生推荐结果

形成当前用户的最近邻居后，就可以利用最近邻居的评分预测出目标用户未评分项目的评分，然后将评分较高的项目作为最后的推荐结果，推荐给目标用户。我们用 P_{ui} 表示用户 u 对项目 i 的评分，则 P_{ui} 可以采用如下策略进行预测

a) 近邻平均评分

该方法选取目标用户 u 的最近邻居对项目 i 的平均评分作为预测值，简单，但效果不好。计算方法如下：

$$P_{u,i}=\frac{1}{|N(u)|}\sum_{u_k\in N(u)}R_{u_k,i} \tag{3-4}$$

b) 相似度加权求和

近邻平均评分没有考虑最近邻居与目标用户的相似度，最近邻集合中的每个用户对预测评分的贡献程度是一样的，为此，相似度加权以最近邻居与目标用户的相似度大小为权重系数，最近邻与目标用户相似度越高，说明他们的兴趣越相似，那么对预测评分的贡献程度就越大。计算方法如下：

$$P_{u,i}=\frac{\sum_{u_k\in N(u)}sim(u,u_k)\cdot R_{u_k,i}}{\sum_{u_k\in N(u)}|sim(u,u_k)|} \tag{3-5}$$

c) 修正的加权求和

相似度加权求和尽管采用了相似度权重，但是没有考虑不同用户的评价尺度问题。修正的加权求和策略通过考虑近邻用户的评分偏差，克服了评价尺度不一致的问题。计算方法如下：

$$P_{u,i}=\overline{R_u}+\frac{\sum_{u_k\in N(u)}sim(u,u_k)\cdot(R_{u_k,i}-\overline{R_{u_k}})}{\sum_{u_k\in N(u)}|sim(u,u_k)|} \tag{3-6}$$

$$\overline{R_u}=\frac{1}{|I_u|}\sum_{i\in I_u}r_{u,i} \tag{3-7}$$

计算完目标用户对未评分项目的评分后，将预测评分由高到低排列，选取前 N 个项目形成 TOP-N 推荐。

3.2.2 基于项目的协同过滤算法

基于用户的协同过滤算法从用户的角度出发，认为用户会喜欢与自己兴趣爱好相似的人喜欢的物品，其现实意义是日常生活中，我们容易接受来自志同道合的人所给出的推荐。和基于用户的协同过滤算法不同的是，基于项目的协同过滤

算法从项目的角度出发，认为用户会选择与自己以前喜欢的物品相关或者类似的物品，其现实意义是我们在购买一个新手机的时候，通常还会购买与之相关的手机套，手机膜，耳机等。总结起来，基于用户的协同过滤算法利用用户与用户之间兴趣相似关系，基于项目的协同过滤算法利用项目和项目之间的相关关系。

基于项目的协同过滤算法是由 Sarwar[40]提出的。Sarwar 针对当时基于用户的电子商务推荐系随着用户规模的快速增长，用户相似度计算需要消耗较长时间，大多数用户只评价了较少商品的缺陷，深入分析电子商务推荐系统中项目间关系与用户间关系差别的基础上，指出项目与项目之间的关系比用户与用户之间的关系更加稳定，从而提出了基于项目的协同过滤算法。我们可以想象，电子商务网站中一天新增加的用户数要远远大于当天新增加的项目数。

与上述基于用户的类似，基于项目的协同过滤算法也分为三步：数据初始化，最近邻居的形成，评分预测及推荐结果的形成。基于项目的协同过滤算法和基于用户的协同过滤算法一样，都是基于用户项目评分矩阵，因此数据初始化过程一样，下面主要介绍最近邻居形成和评分预测及推荐结果的形成。

1) 最近邻居形成

最近邻居的形成基于项目间的相似性。余弦相似性和相关相似性也可以用来计算项目间的相似性，只是计算所用到的用户项目评分向量与基于用户的协同过滤算法不同而已。

a) 余弦相似性

将用户项目评分看成 m 维的列向量，项目间相似性用列向量间的余弦夹角来表示，夹角的余弦值越大，项目间的相似性就越小。设 U_i 和 U_j 为对项目 i 和项目 j 有过评分的用户集合，U_{ij} 为项目 i 和 j 的共同评分用户集合，即 $U_{ij}=U_i\bigcap U_j$，向量 $\vec{i}$ 和 $\vec{j}$ 分别为项目 i 和 j 在 U_{ij} 上的评分，项目 i 和 j 在 m 维项目空间上的评分记为 R_{ui}，R_{vi}，则项目 i 和 j 之间的相似度 $sim(i,j)$为：

$$sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}\cdot\vec{j}}{\left\|\vec{i}\right\|\times\left\|\vec{j}\right\|}=\frac{\sum_{u\in U_{ij}}R_{u,i}R_{v,i}}{\sqrt{\sum_{u\in U_i}R_{u,i}^{\ 2}}\sqrt{\sum_{u\in U_j}R_{u,j}^{\ 2}}} \tag{3-8}$$

b) 修正余弦相似性

记项目 i 和项目 j 的平均评分分别为 $\overline{R_i}$ 和 $\overline{R_j}$，则项目 i 和 j 的修正余弦相似性为

$$sim(i,j)=\frac{\sum_{u\in U_{ij}}(R_{u,i}-\overline{R_u})(R_{u,j}-\overline{R_u})}{\sqrt{\sum_{u\in U_i}(R_{u,i}-\overline{R_u})^2}\sqrt{\sum_{u\in U_j}(R_{u,j}-\overline{R_u})^2}} \tag{3-9}$$

c) Pearson 相关相似性

$\overline{R_i}$ 和 $\overline{R_j}$ 在此处表示项目 i 和项目 j 在两者共同评分用户集合 U_{ij} 上的平均评分。

$$sim(i,j)=\frac{\sum_{u\in U_{ij}}(R_{u,i}-\overline{R_i})(R_{u,j}-\overline{R_j})}{\sqrt{\sum_{u\in U_{ij}}(R_{u,i}-\overline{R_i})^2}\sqrt{\sum_{u\in U_{ij}}(R_{u,j}-\overline{R_j})^2}} \tag{3-10}$$

项目间的两两相似度计算完全后，形成 $n\times n$ 的项目相似度矩阵 $IS(n,n)$，项目相似度矩阵和用户相似度 $US(m,m)$类似，也是对称矩阵，对角线元素表示项目 i 和项目 i 本身的相似性，也置为 0。项目相似度矩阵计算结束后，采用某种邻居选取策略，形成当前项目的最近邻居集 $N(i)=\{i_1,i_2,\ldots,i_k\}$。

2) 评分预测及推荐结果形成

项目的最近邻选取好以后，可以利用最近邻预测出当前用户对未评分项目的评分，最后将预测评分较高的项目，即用户最可能感兴趣的项目，推荐给当前用户。最近邻预测评分同样也有三种计算策略，计算方式分别如下所示：

$$P_{u,i}=\frac{1}{|N(i)|}\sum_{j\in N(i)}R_{u,j} \tag{3-11}$$

$$P_{u,i}=\frac{\sum_{j\in N(i)}sim(i,j)\cdot R_{u,j}}{\sum_{j\in N(i)}|sim(i,j)|} \tag{3-12}$$

$$P_{u,i}=\overline{R_u}+\frac{\sum_{j\in N(i)}sim(i,j)\cdot(R_{u,j}-\overline{R_j})}{\sum_{j\in N(i)}|sim(i,j)|} \tag{3-13}$$

其中 $\overline{R_j}$ 为项目 j 在用户空间上的平均得分。

3.2.3 基于模型的协同过滤算法

基于最近邻的协同过滤算法，包括基于用户的协同过滤算法和基于项目的协同过滤算法，在所有用户或项目空间中搜索最近邻居，随着推荐系统规模的快速发展，用户数越来越多，项目数也越来越多，基于近邻的协同过滤算法的可扩展性和实时性变得越来越差，而基于模型的协同过滤算法根据用户项目评分矩阵所训练出的模型进行推荐，由于模型可以离线预先计算好，推荐系统的可扩展性和实时性大大增强。常用的模型有奇异值分解、聚类模型、云模型等。

(1) 奇异值分解模型(Singular Value Decomposition，SVD)

SVD 是矩阵分解的一种，在某些方面和对称矩阵基于特征向量的对角化类似[43]，能深刻揭示矩阵的内部结构。奇异值分解在图像压缩、信息检索、最小二乘

法和机器学习等方面有着运用广泛。

最近邻模型（最近邻协同过滤算法）和矩阵分解模型是协同过滤算法中最经典的两大模型。个性化推荐的本质就是如何将用户和物品联系起来，最近邻模型通过最近邻来联系用户和物品，矩阵分解则通过隐含特征来联系用户和物品。

矩阵分析模型的基本思想是通过分析用户的行为，将用户的特征和项目的特征用相同数量的因子向量来表示，分别称为用户和项目的隐含特征向量，表示用户和项目对隐含类的关系，然后计算用户特征向量和项目特征向量的内积，如果内积越大，那么说明用户的特征和项目的特征越吻合，也就是用户会更喜欢该项目。

Sarwar 等人最早将奇异值分解引入到推荐系统中，利用矩阵分解将用户项目评分矩阵分解为不同的特征及这些特征对应的重要程度[44]，

矩阵分解模型通过如下公式计算用户 u 对项目 i 的兴趣：

$$preference(u,i)=r_{ui}=p_u^T q_i=\sum_{f=1}^{F} p_{u,k}q_{i,k} \tag{3-14}$$

其中 F 为特征向量，$p_{u,k}$ 表示用户 u 对第 k 个隐含类的关系，$q_{i,k}$ 表示第 k 个隐含类和项目 i 的关系。$p_{u,k}$ 和 $q_{i,k}$ 为待估计参数，通过用户项目评分数据中的训练集数据训练得到。通常采用如下方式计算最合适的 $p_{u,k}$ 和 $q_{i,k}$。

$$\min_{P,Q}\sum_{u}\sum_{i}\frac{1}{2}(r_{ui}-p_u^T q_i)+\lambda(\|p_u\|^2+\|q_i\|^2) \tag{3-15}$$

$\lambda(\|p_u\|^2+\|q_i\|^2)$ 是用来防止过拟合的正则化项，λ 可以通过实验获得。利用最优化理论求解上述最小化问题，便可得到 $p_{u,k}$ 和 $q_{i,k}$，然后将 $p_{u,k}$ 和 $q_{i,k}$ 带入到(3-14)中，便可得到用户 u 对项目 i 的评分。

奇异值分解模型将用户对项目的评分预测问题转化成评分矩阵的参数估计问题，由于评分矩阵的参数个数远远小于用户未评分的个数，极大地缩小了问题的规模，能起到降低矩阵维数的作用，但是降低矩阵维数的同时也会造成信息丢失，影响推荐的准确性。

以下是基于奇异值分解的协同过滤算法：

输入：用户评分矩阵 R

输出：预测评分矩阵 P

Step1：将用评分矩阵 R 规范化为 R_s;

Step2：用奇异值分解的方法分解矩阵 R，得到矩阵 U、S、V;

Step3：取矩阵 S 中的前 K 维，得到 S_k;

Step4：计算 U、V 相应的简化矩阵 U_k，V_k;

Step5：目标用户 *a* 在项目 *i* 上的预测评分为

$$P_{aj} = \overline{R_a} + U_k \times \sqrt{S_k(a)} \sqrt{S_k} V'_k(j)$$

其中 $\overline{R_a}$ 是用户 *a* 的平均评分，U、S、V 分别为 R 经过分解后的左中右矩阵，K 为分解后保留的维数，$U_k \times \sqrt{S_k(a)}$ 为用户矩阵，$\sqrt{S_k} V'_k(j)$ 为项矩阵。

通过奇异值分解，可以将用户评分矩阵从高维空间缩小到低维空间，有效缩小问题的规模。然而由于使用了维数约减技术，丢失了一些信息，会使影响推荐的精度。

(2) 聚类模型

聚类分析是数据挖掘和机器学习中的一种重要算法，属于无监督学习。聚类是根据对象的特征将对象进行分类的一种技术，使得高度相似的对象成为一类，同时不同类之间的具有高度的相异性。聚类分析的应用非常广泛，在电子商务、市场分析、生物学、Web 挖掘、空间数据处理和卫星照片分析等方面有着广泛的应用。

基于内存的协同过滤算法需要在整个用户或项目空间上查找目标用户的最近邻居。随着用户数和项目数飞速增长，推荐系统需要同时为海量用户提供实时推荐服务，如果还是在整个用户或项目空间上查找最近邻居，势必会对推荐系统的实时性造成较大影响，使得推荐系统无法追踪用户的兴趣喜好，降低了用户体验。

如果利用聚类技术，对用户或者项目进行聚类，那么只需寻找聚类中心就可以快速找到最近邻居，不必在整个项目空间中去寻找最近邻居了，大大缩短了最近邻居查找的时间，这就是基于聚类的协同过滤算法的基本思想。因此在协同过滤算法中采用聚类分析，将用户评分矩阵划分成相似度较高，数据规模较小的矩阵，既能减少所需要处理的问题规模，还可以减少评分稀疏性的影响。

现有的基于聚类的协同过滤算法中主要的聚类模型有用户聚类模型、项目聚类模型和用户项目联合聚类模型。用户聚类模型首先对用户聚类，使得相似度较高的用户聚成一类，同时产生若干个聚类中心，然后根据目标用户与各个聚类中心之间的相似性及目标用户所属类别矩阵，选择当前用户的最近邻居[45]。项目聚类首先根据用户评分对项目进行聚类，将用户评分相似的项目放在一个聚类中[46]。无论是基于用户还是基于项目的聚类，都只考虑了用户或项目单方面的关系，未考虑用户和项目之间的相关性[47]。为此产生了基于用户和项目的联合聚类方法。联合聚类是一种对存在行列相关性的矩阵进行聚类的重要方法，基本原理就

是对行聚类和对列聚类，两个步骤循环迭代直至收敛[48]。

聚类是一种将具有相似属性的对象聚集在一起的一种无监督学习技术，受到数据挖掘和机器学习领域的广泛研究。目前已经提出了很多聚类算法，比如K-means、K-medoids、CLARANS、BIRCH、DBSCAN 等算法[49]。以用户聚类为例，聚类方法选取最为著名的 K-means，基于用户的 K-means 聚类算法的基本步骤是：

Step1：设 K 为聚类中心的个数，从用户项目评分矩阵中选取评分最多的 K 个用户作为初始的聚类中心，记为$\{u_1,u_2,\ldots,u_k\}$。

Step2：计算用户集中的每个用户 u 与各个聚类中心的的相似度 $Sim(u,u_i)$，将用户 u 添加加到与之最相似的类中

$C_i=\max_i sim(u,u_i)$。

Step3：计算每个聚类的均值，作为新的聚类中心。

Step4：如果聚类中心未发生改变或者算法迭代次数达到设定的值，则算法停止迭代，得到最新的 K 个聚类，否则跳转到 Step2 继续迭代。

(3) 云模型

云模型是由李德毅院士提出的，能够实现定性概念及其值表示间的转换[50]，广泛运用在智能控制和模糊评测等方面。云模型通过云的 3 个数字特征来表达概念的整体特征，他们是期望 Ex（expected value）、熵 En（entropy）、超熵 He（hyper entropy）。期望是最能代表定性概念的点，熵表示定性概念的不确定性度量，取决于概念的随机性和模糊性，超熵是熵的不确定性度量，由熵的不确定性和模糊性来共同决定，期望、熵和超熵能够表示定性概念的整体特征，记作 $C=(Ex,En,He)$，称为云的特征向量。

设用户 u 的评分集合 $P_u=\{u_1,u_2,\ldots,u_N\}$，用户 v 的评分集合为 $Pv=\{v_1,v_2,\ldots,v_M\}$，其中 N 和 M 为用户 u 和 v 的评分项目数，则用户 u 的评分向量的样本均值为

$$\overline{X}=\frac{1}{N}\sum_{i=1}^{N}x_i$$

一阶样本绝对中心矩为

$$\frac{1}{N}\sum_{i=1}^{N}\left|x_i-\overline{X}\right|$$

样本方差

$$S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{X})^2$$

Ex_i 的估计值为

$$E_{\hat{x}} = \overline{X}$$

He_i 的估计值

$$He = \sqrt{\frac{\pi}{2}} \times \frac{1}{N}\sum_{i=1}^{N}\left|x_i - \overline{X}\right|$$

En_i 的估计值为

$$En = \sqrt{s^2 - \frac{1}{3}He^2}$$

用户 u 和 v 的云向量为 Cu=（Ex_u,En_u,He_u），Cv=（Ex_v,En_v,He_v），用户 u 和用户 v 的云相似度为云向量间的夹角

$$YSim = cos(\mathrm{C}_i, \mathrm{C}_j) = \frac{C_i \square C_j}{|C_i||C_j|}$$

利用上述方法，可以计算用户评分矩阵中用户间的两两相似度，形成用户相似度矩阵。获得用户相似度矩阵后，就可以按照基于内存的协同过滤算法中近邻选择和评分预测的方法来预测目标用户对未评分项目的评分。

3.3 协同过滤算法的改进研究综述

现有的协同过滤算法研究综述大多是从协同过滤算法所存在问题的角度进行分析，比如稀疏性和冷启动问题，忽视了协同过滤算法本身，重现象轻本质，这种本末倒置的认识方法不利于认清协同过滤算法中根本问题产生的原因。为此，本文从协同过滤算法本身出发，以协同过滤算法的三个步骤为角度，综述了近期协同过滤算法国内外的研究进展。

3.3.1 相似度改进

协同过滤算法最开始计算的是相似度，无论是后面的邻居选择还是评分预测，都需要利用第一步计算出的相似度，因此相似度计算是协同过滤算法中最为关键的一步。

协同过滤算法中传统的相似度计算方法主要有余弦相似度、修正余弦相似度和相关相似性。总体角度来讲，这三种相似度计算方法只有在用户存在共同评分项目时，才能计算相似度，否则两个用户间的相似度为 0。并且电子商务网站中，用户购买的商品本来就很少，往往不到所有商品数量的 1%，用户间共同购买的

商品个数就更加少之又少了。用户间的相似度是一种客观存在，两个用户即使没有购买同一个商品或者对同一个项目进行过评分，但他们的兴趣爱好同样也会存在一定的相似度，因此在传统相似度计算方法的前提条件下，为 0 的相似度掩盖了客观存在的相似度。

三种相似度分开来看，余弦相似度将用户评分看作向量，通过计算用户评分向量夹角的余弦值来反映用户间的相似度。这种计算方法存在几点不足，一是评分向量中未包含用户评分的统计特征，每个用户的评分习惯是不一样的，有的人喜欢打高分，有的喜欢打低分。二是如果用户间的评分向量平行，也就是夹角为 0，那么余弦相似度为 0，而事实上用户间的相似度应该很高。

修正余弦相似度在余弦相似度的基础上考虑了用户评分的统计特征，但该方法更多反映的是用户间的相关性而非相似性，相关性和相似性是两个不同的概念，前者反映组合，而后者反映聚合。

相关相似性依据共同评分项目计算相似度。共同评分项目上的评分确实能够很好地反映用户间的相似度，但是在用户项目评分矩阵变得稀疏的情况下，用户共同评分的项目往往很少，相关相似性的准确性受到较大影响，因此相关相似性受数据稀疏性的影响较大。比如当用户间不存在共同评分的项目时，相关相似性变得无法计算。当用户间只有 1 个共同评分项目时，不管他们对项目的评分是多少，相关相似性均为 1。此外如果用户的平均评分和用户对某一共同评分项目的评分相同，Pearson 计算公式的分母为 0，相关相似性此时也无法计算。

协同过滤算法一直饱受稀疏性问题和冷启动问题的困扰。稀疏问题导致用户与用户之间由于没有共同的评分项目而无法计算相似度。冷启动问题是新用户或者新产品进入到推荐系统，系统中没有任何关于他们的记录，导致无法为其推荐喜欢的产品或者为其找到喜欢的用户，可见冷启动问题是稀疏问题的极端。

由上述分析可知，在共同评分项目上计算相似度是导致协同过滤算法受稀疏性问题和冷启动问题困扰的根本原因。针对该问题，研究人员提出了很多新的相似度度量方法，避免了在共同评分项目的基础上计算相似度。

许鹏远等提出了元相似度的概念[51]，然后根据元相似度的概念，提出了相应的改进算法。元相似度借鉴了元数据的概念，元数据是关于数据的数据，元相似度则是关于相似度的相似度。由于元相似度是基于相似度矩阵计算而来的，因此即使两个用户未购买同一产品，即两个用户不相关，那么元相似也可以计算这两个用户的相似度。元相似度的基本思想是如果两个人和第三个人的相似性都很高，那么这两个用户也应该很相似，也就是说两个用户相似的人越多，这两个用户也就越相似。如果从网络的拓扑结构来理解，传统相似度反映的是局部的直接的相

似关系，元相似度则反映的是全局的间接的相似关系。

Panagiotis[52]针对传统相似度计算方法依赖于共同评分项目集，即两个用户评分项目集的交集，提出了 Union 相似度计算方法，在用户评分项目集并集的基础上计算相似度，能有效缓解数据稀疏性问题。后来李聪等人[53]在 Union 相似度的基础上进行了深入分析，将用户评分项目并集中的用户根据有无推荐能力分为两种类型，对于前者不再计算相似度，对于后者使用领域最近邻对评分项目并集中的未评分项进行评分预测，提高了最近邻查找的准确性。

此外，研究人员还提出了其他的相似度改进相似度，比如基于熵的相似度[54]、PIP 相似度[55]、有向相似性[56]、结构相似性[57,58]等。

除了提出新的更准确的相似度，避免在共同评分项目上的计算相似度外，近来还出现了一些克服传统相似度计算的不足的新方法。赵琴琴等人[59]利用传播的思想对协同过滤算法中的稀疏问题进行研究，提出了基于内存的传播式协同过滤算法 SPCF(Similarity Propagation based Collaborative filtering)，SPCF 首先利用传统相似度计算用户之间的相似度，然后通过 SimTrans 算法[60]在用户相似度矩阵中传播相似度，以帮助目标用户找到更多更可靠的邻居。受社会网络中信任关系具有传播性质的启发，李琳娜等人[61]认为物品之间的相似性也可以传播，提出了基于启发式的物品相似度传播协同过滤推荐方法，该算法建立物品相似网，对通过相似网相连的两个物品重新计算相似度。

3.3.2 邻居选择改进

最近邻选择是协同过滤算法的第二步。相似度计算的目的之一就是为了选出最近邻居，评分预测主要就是利用最近邻的评分，可见最近邻选择在协同过滤算法中起到承上启下的作用，因此邻居选取会对协同过滤算法的推荐效果产生较大的影响。

常见的邻居选取策略主要有[62]：

1) 基本策略：选取与目标用户相似度最高的前 K 个用户作为邻居，并且这 K 个用户还对目标项目有过评分。

2) 带重叠度阈值的基本策略：选取与当前用户相似度最高的前 K 个用户，这 K 个用户不仅对目标项目有过评分，而且与当前用户的评分数不小于制定阈值。

3) 相似性策略：挑选相似度最高的 K 个用户，不管 K 个用户有没有对目标项目有过评分。

4) 联合策略：结合基本策略和相似性策略分别选取一定数量的邻居。

5) 带重叠度阈值的联合策略：用基本策略和相似性策略在选取邻居时，还要求这些用户与当前用户的共同评分数不小于指定阈值。

上述的各种邻居选取策略，均有所不足。相似性策略依据相似性选取最近邻居，虽然能够保证选取的最近邻居与目标用户有较高的相似性，但是这些最近邻居中对目标用户有过评分的却很少，因而这些较高相似度的最近邻在后面的评分预测中完全发挥不了任何作用。基本策略选取的是对目标用户有过评分的最近邻，但是却会选取相似度较低的最近邻，也就是会选取一些与当前用户兴趣不是很一致的用户当作了最近邻，这势必会降低推荐结果的准确性。

稀疏性问题是协同过滤算法中的研究重点，但是现有的研究广泛关注于相似性计算阶段的稀疏性，其实邻居选择阶段也存在稀疏性问题。相似性策略选取的最近邻不一定会对目标用户有过评分，那么这些未评分的最近邻通常会被过滤掉，即使没有过滤掉，他们在评分预测阶段也不会对推荐产生任何作用，其实也相当于被过滤掉了。由上述分析可知，近邻选择阶段也存在稀疏性问题，而且要比相似性阶段更加隐蔽。近邻评分稀疏会使得参加预测的邻居数量大大减少，甚至导致协同过滤算法无法对某些项目做出评分预测，可见近邻评分稀疏会降低协同过滤算法的准确性和覆盖率。

为此研究人员利用填充和传播的思想来改善近邻评分稀疏的问题。Zhang 等人[62]借鉴递归算法的思想，提出了基于递归的协同过滤算法，该算法利用协同过滤算法预测出未评分的邻居对目标项目的评分，然后在评分预测阶段利用预测出的评分计算目标用户对目标项目的评分。冷亚军等人[63]利用 SVD 填充近邻的缺失评分，提出了近邻评分填补的协同过滤算法。现实世界中，朋友的朋友也会和我们有相同的兴趣，他们的建议对我们也有一定的帮助。为此，宣照国等人[64]借鉴该思想，利用传播的方法，提出了基于扩展邻居的协同过滤算法，该算法在相似度计算的基础上，通过扩展目标用户最近邻，让邻居的邻居也参与到推荐中来。

除了近邻评分稀疏外，还有其他的一些改进的近邻选择方法。贾冬燕等人[65]针对协同过滤算法中的准确性和抗攻击性问题，根据传统相似度方法计算出的相似度和用户信任计算模型计算出的信任度，提出了基于双重邻居选取策略的协同过滤算法。Huete 等人[66]基于如果最近邻能够预测目标用户过去的评分，那么他们也可以预测目标用户将来的评分的假设，提出以邻居用户预测过去评分的准确性来选择最近邻。Bellog ń 等人[67]利用图划分的方法进行谱聚类，从而寻找到当前用户的最近邻。

3.3.3 评分预测改进

协同过滤算法的目的就是为用户提供个性化推荐，一般会将预测评分最高的前若干个项目推荐给目标用户，因为预测评分越高，表明用户喜欢该项目的可能性就越大。无论是相似度计算还是邻居选择，均是为评分预测阶段提供数据准备，

都是在为评分预测做铺垫，可见评分预测是协同过滤算法中起到决定性的作用。

标准的评分预测方法以目标用户历史评分的平均值为基准，以邻居的相似度加权邻居对目标项目的评分来逼近目标用户对目标项目的评分。该种预测策略过高估计了历史评分对预测的影响。历史评分高并不能代表用户就对该项目感兴趣，历史评分低的用户也可能对该项目感兴趣。针对该问题，陈志敏等人[68]提出了基于相关均值的协同过滤算法，利用相关均值来代替历史评分均值，更准确地反映了目标用户对当前项目的真实偏好。

该种策略除了高估历史评分外，还低估了邻居用户在推荐过程中的作用，因为邻居用户的相似度加权平均评分通常都比用户的评分评分要小，也就是说邻居用户的评分主要是由用户的平均评分来决定的，因此如何充分发挥邻居在推荐过程中的作用是值得研究的一个方向。

协同过滤算法产生的预测评分通常是小数，但是实际的推荐系统中评分通常为正整数，比如 MovieLens 采用五分制，为 1-5 分，EachMovie 为 1-10 分，传统算法通常按照四舍五入的方法对预测评分进行判定，未考虑用户的评分偏好，因为有的用户喜欢打高分，有的用户喜欢打低分。针对该问题，李永等人[69]提出根据趋势度、偏离度和判定度综合判断预测评分，实验结果表明该方法可以提高推荐算法的准确性，但是对于每个用户的评分都计算趋势度、偏离度和判定度，计算会相当复杂，后来李春等人[70]提出简化方法，直接根据用户的评分趋势来对预测评分进行取整。

3.4 本章小节

本章首先介绍了协同过滤算法的概念、原理及基本假设，然后分步骤阐述了基于用户和基于项目的协同过滤算法，介绍了标准的相似度计算方法，常用的邻居选取和评分策略，而后介绍了基于模型的协同过滤算法，包括 SVD 模型、聚类模型和云模型，接着给出了协同过滤算法的分类，最后从协同过滤算法的三个步骤出发，分别分析了相似度计算、最近邻选择和评分预测各个步骤所存在的问题，同时综述了相应的改进策略，为认识协同过滤算法和后续改进协同过滤算法提供了新的思路。

第四章 考虑负相关性信息的协同过滤算法

传统协同过滤算法利用 Pearson 相关系数计算用户或项目相似度，采用 K 近邻策略选取与当前用户或项目相似度最高的前 K 个用户或项目作为最近邻居，仅考虑了 Pearson 相关系数的正相关性，未考虑其负相关性。针对该问题，本章提出了考虑负相关性信息的协同过滤算法，其中，最近邻居由当前用户的正相关用户组成，最远邻居由当前用户的负相关用户组成，使用可调节的参数控制最近邻居和最远邻居在推荐过程中的作用。数值实验表明，负相关性信息不仅可以提高推荐结果的准确性还可以增加推荐列表的多样性。此外，我们还发现负相关性信息能大幅度提高度小用户的推荐准确性。因此负相关性信息有助于解决推荐系统中的冷启动问题和同时保证准确性和多样性的问题。

4.1 问题描述

个性化推荐算法利用用户的历史选择信息预测其喜好，成为解决信息过载的有效手段之一。目前已经提出了许多推荐算法，比如协同过滤算法[71]、基于内容的推荐算法[72]、混合推荐算法[72]和基于网络结构的推荐算法[2]等，其中协同过滤算法是运用最为广泛和成功的个性化推荐算法。在基于协同过滤的推荐系统当中，K 近邻模型是使用最广泛的经典模型[73]，其主要原理是利用评分相似度构造 K 个最近邻居，然后根据最近邻对用户进行推荐。

传统协同过滤算法常常利用 Pearson 相关系数[72]计算相似度，根据相似度由大到小选择 K 个用户组成最近邻，该算法有一个明显的不足就是只考虑 Pearson 相关系数的正相关性，忽略了其负相关性。某些文献虽然考虑到了负相关性，但是都没能表示出负相关性的真实含义，比如认为负相关性对推荐结果影响不大，进而将其视为无效值直接舍弃掉[74]或使用绝对值来代替[71]或将其投影成正相关性[75]。因此为了研究负相关性信息对协同过滤算法的影响，本章提出了考虑负相关性信息的协同过滤算法(Collaborative Filtering by Considering Negative Correlation, CNCF)，该算法可以同时考虑用户的正负相关用户。

4.2 相关研究综述

已有的对协同过滤算法的改进主要集中在数据稀疏性问题和相似性度量问题。针对稀疏性问题，研究者提出了很多解决方法，其中包括矩阵填充[18]和矩阵降维[76]，邓爱林等人[18]等提出基于项目评分预测的协同过滤算法，把根据项目相似性预测出的用户对未评分项目的评分填充到用户项目评分矩阵中，从而在相对稠密的用户项目评分矩阵上进行协同过滤。孙小华等人[76]等利用奇异值分解(SVD)

对项目评分矩阵进行降维，将高维稀疏评分矩阵投影到低维，有效缩小问题规模。关于相似性度量的已有研究包括有向相似性[56]和项目间相似性[77]等。

本文主要从邻居选择的角度对协同过滤算法进行改进。邻居选择的研究主要有以下几个方面，如罗辛等人[78]提出相似度支持度的概念并将其作为一种新的 K 近邻度量。黄创光等人[79]提出不确定近邻的协同过滤算法，自适应地选择预测目标的近邻对象作为推荐群。张尧等人[80]综合考虑邻居选择权重，提出了基于用户分类的邻居选择方法。Zeng 等人[81]研究了不相似用户对传统协同过滤算法的影响。当用户评分数据相当稀疏或者分布不均衡时，Pearson 相似度值可能为负值，现有文献往往使用其绝对值[71]或直接舍弃掉[74]。针对负相关性，Wang 等人[75]将负相似度映射到区间[0,1)上，正相似度映射到区间(0.5,1]上。不论是使用绝对值还是映射到指定区间，均没有考虑到 Pearson 负相似度信息。

综上所述，关于邻居选择的现有研究，较少关注到 Pearson 相关系数的负相关性，要么直接舍弃掉[74]，要么使用绝对值代替[71]，要么投影成正相似度[75]，这些处理方式，均没有考虑到 Pearson 负相关性的真实含义。为此本文提出了考虑 Pearson 负相关性信息的改进协同过滤算法，从负相关性的真实含义出发，将负相关用户集成到协同过滤算法中，和正相关用户一起产生推荐。

4.3 传统协同过滤算法的不足

协同过滤算法的主要思想是利用目标用户的相似邻居来对用户进行推荐，因此邻居用户的选取对算法准确性和性能至关重要。基于 Pearson 的最近邻协同过滤算法在选取邻居时，忽视了 Pearson 相关系数的两面性，只考虑 Pearson 相关系数的正相关性，没有考虑 Pearson 相关系数的负相关性，下面用一个实例来更好地说明该问题。

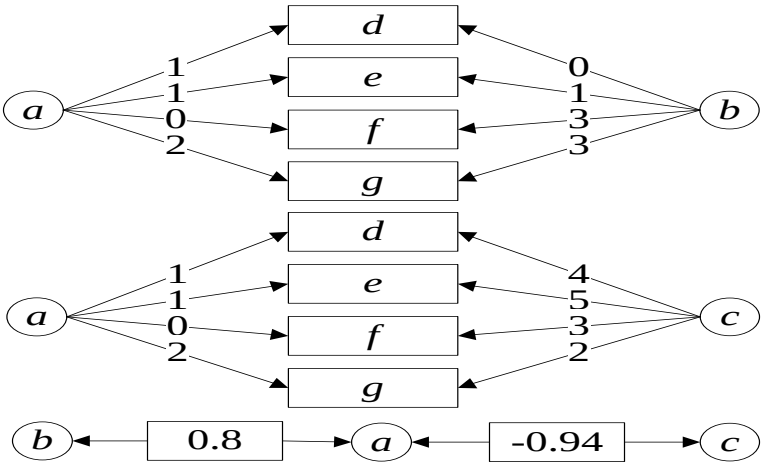


图 4-1　用户评分示意图

图 4-1 表示用户 *a*，*b*，*c* 对项目 *d*，*e*，*f*，*g* 的评分。采用 Pearson 相似度计算

图 4-1 中用户 a、b 和 c 之间的两两相似度，其中用户 a 和其他用户间的相似度分别为：Sim(a,b) = 0.8，Sim(a,c) = -0.9，由（3-3）式可知，Pearson 相关系数 $r \in [-1,1]$，$r > 0$ 表明用户在共同评分项目上的评分是正相关的，如图 4-1 中，Sim(a,b)=0.8，a 和 b 在共同评分项目上的评分正相关，即一方对某个项目给高分（低分）时，另一方也倾向于给高分（低分），因此我们可以利用用户 b 对项目 f 的评分来预测用户 a 对项目 f 的评分。同理，$r < 0$ 表示用户在共同评分项目上的评分是负相关的，如图 4-1 中，Sim(a,c)=-0.94，用户 a 和用户 c 在共同评分项目上的评分负相关，即一方对某个项目给高分（低分）时，另一方倾向于给低分（高分），因此我们同样可以利用用户 c 对项目 f 的评分来预测用户 a 对项目 f 的评分。

由上述分析可知，基于 Pearson 的最近邻协同过滤算法未充分利用负相关性信息，为此，本文在选择邻居用户时，从两个方面同时考虑正负相关两方面的信息。

4.4 考虑负相关性信息的协同过滤算法

4.4.1 邻居选取

最远邻居（furthest neighbor）：与当前用户的评分成负相关性的用户集合，即负相关用户集合。

$$FN(u_i) = \{u \mid Sim(u_i, u_j) < 0\}, i \neq j \tag{4-1}$$

其中 $FN(u_i)$为 u_i 的最远邻居集合，$Sim(u_i,u_j)$为用户 u_i 和 u_j 的 Pearson 相似度值。

K-最远邻居集合（K- furthest neighbor）：从最远邻居集中选择前 K 个绝对值最大的用户。

$$KFN(u_i) = \{u_1, u_2, ..., u_k\}, u_i \notin KFN(u_i) \tag{4-2}$$

其中用户 $u_j (1 \le j \le k)$ 按与 u_i 的相似度的绝对值由大到小排列。

4.4.2 评分预测

本文将最远邻居集成到协同过滤算法中，故采用如下公式进行评分预测。

$$P_{u_t,i} = \overline{R_{u_t}} + (1-\alpha)\frac{\sum_{v \in N_{u_t}} sim(u_t, v) \times (R_{v,i} - \overline{R_v})}{\sum_{v \in N_{u_t}} (|sim(u_t, v)|)} + \alpha \frac{\sum_{w \in FN_{u_t}} sim(u_t, w) \times (R_{w,i} - \overline{R_w})}{\sum_{w \in FN_{u_t}} (|sim(u_t, w)|)} \tag{4-3}$$

其中 FN_{u_t} 为用户 u_t 的最远邻居，α 为阈值，用于调节最近邻居和最远邻居的作用，当 α＝0 时，推荐完全按照最近邻居进行，退化成传统协同过滤算法；当 α＝1 时，推荐完全根据最远邻居进行，α 取值范围为 $\alpha \in [0,1]$。

4.5 实验过程与结果分析

4.5.1 数据集

本文使用 MovieLens 数据集来测试改进后的算法，该数据集是由 GroupLens 研究小组提供的一个著名电影评分数据集，记录了 943 位用户对 1682 部电影的 10 万条打分（评分值为 1~5 之间的整数）。本实验将数据集按照 80%和 20%的比例划分成训练集和测试集。

4.5.2 评价标准

1) 平均绝对误差（Mean Absolute Error）[82]

本文算法的准确性根据平均绝对误差来衡量。MAE 通过计算预测的用户评分与实际用户评分之间的偏差来度量预测的准确性。预测的用户评分集合为 $\{p_1, p_1, ..., p_n\}$，相应的实际用户评分集合为$\{q_1, q_1, ..., q_n\}$，则 MAE 通常定义为

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

MAE 的值越小，表明预测出的评分和实际评分之间的偏差越小，也就是算法准确性越好。

2) 平均 Hamming 距离（Average Hamming Distance）[82]

推荐列表的多样性根据平均 Hamming 距离度量。用户 u_i 和用户 u_j 的推荐列表的多样性被定义为

$$H_{ij} = 1 - \frac{Q_{ij}}{L} \tag{4-5}$$

其中 L 为推荐列表的长度，Q_{ij} 为用户 u_i 和用户 u_j 推荐列表交集中的项目个数。取所用用户对间的 H_{ij} 的平均值<H>作为整个推荐列表的多样性，可见<H>越大，推荐列表的多样性越好。

4.6 实验结果及分析

4.6.1 Pearson 相似度值分布

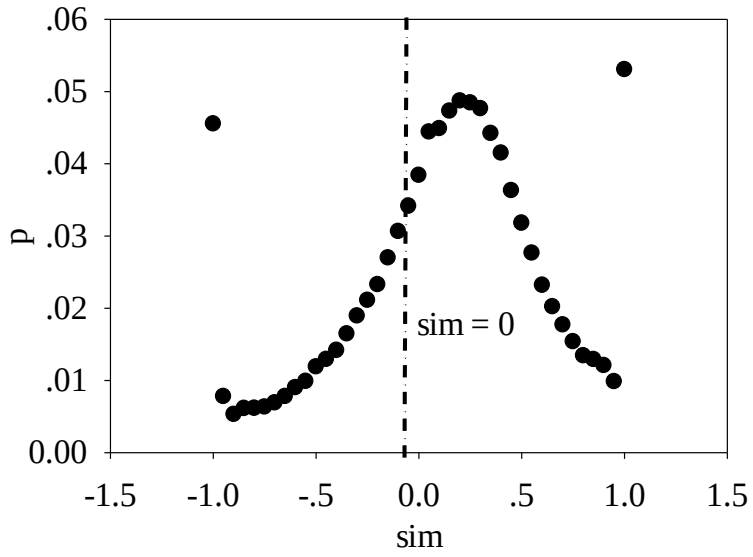


图 4-2 Pearson 相似度值分布情况

本文采用公式(3-3)计算训练集中用户间的两两相似度，相似度值的分布如图 4-2。负相似度值在协同过滤算法中未得到充分利用，如果用户相似度矩阵中负相似度值所占的比例较少，即使未得到利用，最后对结果的影响也不会很大，然而从图 4-2 中我们可以看到，用户相似度矩阵中有 35%的相似度值小于 0，超过了总数的三分之一，数量还是相当多的，因此我们不能随意舍弃掉这些为数并不少的负相似度值。

4.6.2 参数 α 估计

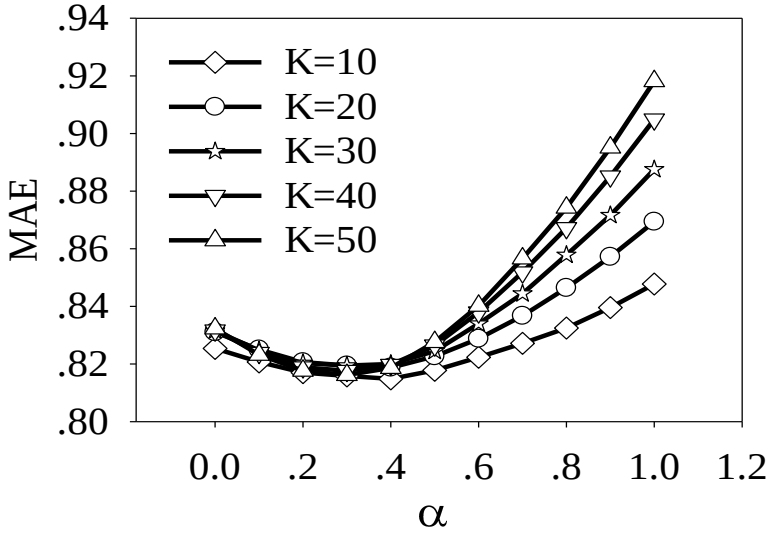


图 4-3 参数 α 估计

考虑负相关性信息的协同过滤算法有邻居个数 k 和用于调节最近邻居和最远

邻居作用的阈值 α 两个参数，本实验通过设置不同的 k 值和 α 值，估计到一个较优的参数值 α，然后应用到本章接下来的其他实验当中。本实验根据不同的邻居个数 k 和阈值 α，分别计算 MAE，邻居个数 k 在 10 到 50 之间变动，阈值 α 在 0 到 1 之间变动，实验结果如图 3 所示。由图 4-3 可知，在不同的邻居个数 k 下，α ＝0.3 时，MAE 都具有最小值，由此可知在该数据集下 α＝0.3 时 CBCF 算法的准确性最好，因此我们选择参数 α=0.3 进行后续实验。

4.6.3 准确性比较

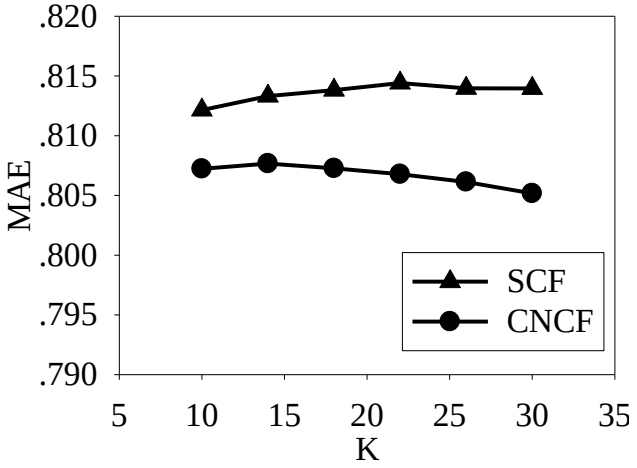


图 4-4 算法准确性比较

为了检验本文提出算法的准确性，我们在同等数据集的基础上，变换邻居个数，比较 CBCF 和基于 Pearson 相似度的传统协同过滤算法（Standard Collaborative Filtering, SCF），计算 MAE，邻居个数 k 从 10 到 30，间隔为 4。由图 4-4 可知，在各种实验条件下，与传统的协同过滤算法（SCF）相比，本文提出的考虑负相关性信息的协同过滤算法（CNCF）均具有较小 MAE。由此可知，与 SCF 相比，CNCF 能明显提高评分预测的准确性，因此我们可以说负相关性信息可以明显提高算法的准确性。

4.6.4 多样性比较

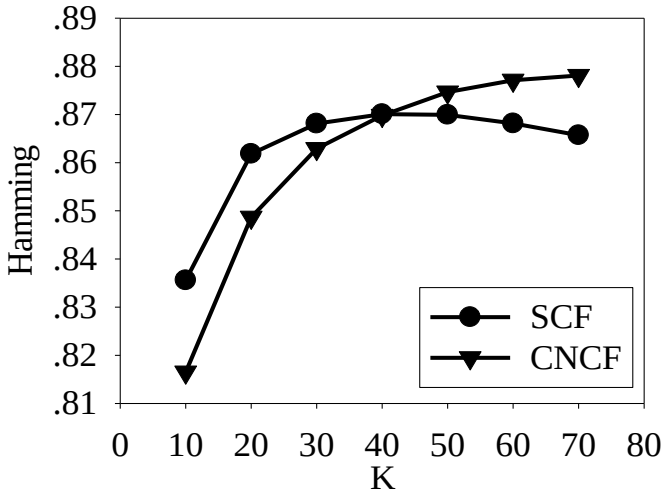


图 4-5 算法多样性比较

为了检验本文提出算法的多样性，我们将本文提出的算法 CNCF 和 SCF 进行比较，以平均 Hamming 距离作为评价指标，邻居个数在 10 到 70 间变动，推荐列表长度 $L=50$。由图 4-5 可知，当邻居个数 k 大于 40 时，CNCF 具有较大的平均 Hamming 距离。随着领居个数的增加，CNCF 算法的多样性逐渐上升，SCF 却呈下降趋势，CNCF 的提升幅度越来越大，这表明负相关性信息可以提高算法的多样性。

给用户推荐流行产品，可以提高准确性，但是会让用户的视野变得狭窄。给用户推荐一个冷门产品或者打分很低的产品，可以提高推荐的多样性，但是很容易引起用户的反感，准确性和多样性之间存在竞争关系，通常只能两者求其一，很难达到平衡[4]。然而本文发现，与 SCF 相比，CNCF 利用了负相关性信息，同时提高推荐的准确性和多样性。

4.6.5 负相关性对度大度小用户的影响

用户的度表示该用户选择过多少产品[2]。为了验证负相关性对度大度小用户的影响，分别计算训练集中度最大的前 100 个用户和度最小的前 100 个用户的 MAE。由 4-6 (*a*)和(*b*)可知，与度大用户相比，考虑负相关性信息的协同过滤算法的准确性对度小用户的提升幅度更大，比如当 $k=16$ 时，度小用户的提升比例可以达到 3.01%，而度大用户的提升比例仅为 0.31%，这表明负相关性可以提高协同过滤算法对度小用户的预测准确性。新用户由于没有打分信息或者有很少的打分信息（即度小用户），协同过滤算法无法为其产生推荐，这就是一直困扰推荐系统领域的冷启动问题[4]。然而本文发现与传统协同过滤算法相比，考虑负相关性信息的协同过滤算法由于将负相关用户集成到推荐过程中，能大幅度提高对度小用户的

推荐准确性，因此负相关性信息有助于解决推荐系统中的冷启动问题。

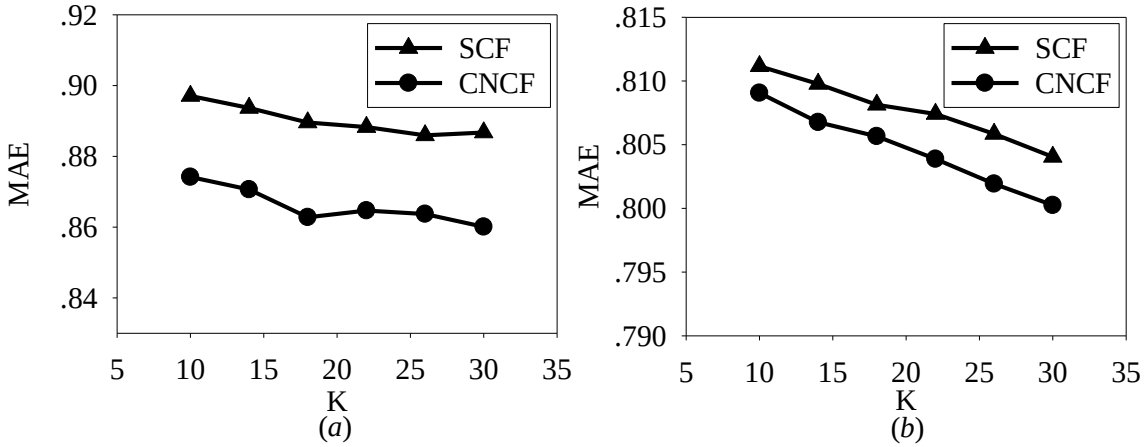


图 4-6 (*a*)为算法对训练集中度最小的前 100 个用户的影响，(*b*)为算法对训练集中度最大的前 100 个用户的影响。

4.7 本章小结

本文深入研究了负相关性信息对协同过滤算法的影响，提出了考虑负相关性信息的协同过滤算法，该算法选取正相关用户和负相关用户分别作为最近和最远邻居，然后使用参数调节两者在推荐过程中的作用，然后预测出用户对未评分项目的评分，最后选取评分较高的产品组成推荐列表。MovieLens 实验表明负相关性信息可以同时提高推荐准确性和推荐列表多样性，这表明负相关性有助于解决同时保证准确性和多样性的问题。进一步分析发现，与度大用户相比，负相关性信息能大幅度提高度小用户的推荐准确度。与传统协同过滤算法相比，本文算法对度小用户的 MAE 提高比例可以达到 3.01%，这表明负相关性信息有助于解决冷启动问题。

第五章 基于 Sigmoid 权重相似度的协同过滤算法

为了解决传统协同过滤算法在稀疏数据条件下相似度计算不准确无法发现有效最近邻问题，提出了基于 Sigmoid 权重相似性的协同过滤算法。首先计算用户间的共同评分次数，然后使用经 Sigmoid 函数调整后的共同评分数加权相似度，产生更准确有效的最近邻。MovieLens 实验表明该算法不仅能获得比传统协同过滤算法更好的预测准确性和推荐覆盖率，而且能弥补权重相似度手动调节参数的不足。进一步分析发现该算法还能提高度小用户的预测准确性。该工作表明 Sigmoid 权重相似度能有效缓解数据稀疏性问题和冷启动问题。

5.1 问题描述

近年来推荐系统能帮助用户克服信息过载带来的负面影响，被广泛成功地运用到各大电子商务网站，例如 Amazon、淘宝和京东等电子商务网站都在使用推荐系统。协同过滤算法是推荐系统中应用最成功的一种个性化推荐技术，典型的协同过滤是基于用户的协同过滤，其基本原理是利用评分相似度构造目标用户的兴趣最近邻，根据兴趣最近邻向目标用户进行推荐[2]。

传统协同过滤算法在用户共同评分项目集上计算相似度，随着电子商务网站快速发展，用户在项目空间上的评分变得稀疏，用户共同评分变得少之又少，导致传统相似度不能准确反映用户兴趣相似关系，降低了目标用户最近邻的准确性和有效性，严重损害了推荐系统的精度。传统协同过滤算法依据共同评分计算相似度，但没有考虑共同评分项目集的大小，改进的权重相似度[83,84]虽然考虑到了共同评分项目集的大小，可一方面引入需要手动调节的参数，该参数容易受数据集的影响，不同数据集会有不同的最优参数，极大限制了方法的使用，另一方面只考虑降低由较小共同评分计算而来的相似度，未考虑较大共同评分的情况。为此本文提出基于 Sigmoid 权重相似性的协同过滤算法(Collaborative Filtering Based on Sigmoid Weight Similarity, SWCF)，该算法既考虑共同评分项目集的大小，又弥补权重相似度上述两方面的不足。

5.2 相关研究综述

相似性度量是协同过滤算法的核心，在算法中起到承上启下的作用，既是邻居选择阶段近邻选择的标准，又是评分预测阶段近邻用户的权重系数，因此国内外学者对准确度量相似性做了大量工作，提出了许多计算方法，主要有 Pearson[85]、Cosine [40]、修正 Cosine [40]和 Jaccard[86]相似性等，其中 Pearson 是目前协同过滤算法中最常见的相似性度量方法之一。这些传统相似度计算方法虽然在共同评分项

目上计算相似度，但没有考虑共同评分项目集的大小。随着电子商务网站系统规模的快速增长，评分数据极端稀疏，用户共同评分数通常只有 1 到 2 个。针对该问题，Herlocker 等人[83]最早提出了 Max 权重相似度，利用参数阈值修正相似度值，有效降低稀疏数据下共同评分较少但评分很相似用户间的相似度值，但是当用户间共同评分数大于参数阈值时，Max 权重会产生修正后的相似度大于 1 的不合理现象，为此 McLaughlin 等人[84]提出了 Min 权重相似度，当用户共同评分数大于参数阈值时不修正相似度，有效弥补了 Max 权重相似度值的缺陷。但是 Max 和 Min 权重相似度分段不连续地修正相似度值，为此 Bell 等人[87]提出连续修正的 Shrink 权重相似度。

综上所述，传统相似度没有考虑共同评分大小，后续的各种权重相似度虽然考虑了，但一方面引入手动调节参数，需要交叉验证才能确定最优值，极大限制了方法的使用，另一方面没有从整体角度出发，只降低共同评分数较小的相似度，没有修正较大的情况，因为用户共同评分项目数越多，用户的兴趣相似的可能性越大。为此本文提出基于 Sigmoid 权重相似度的协同过滤算法，克服了传统相似度和权重相似度的上述弊端。

5.3 基于 Sigmoid 权重相似度的协同过滤算法

5.3.1 传统相似度和权重相似度的不足

传统相似度在共同评分项目集上计算相似度，没有考虑共同评分项目集的大小，使得共同评分项目集较小但评分相似性较高的用户成为当前用户的最近邻居，而共同评分项目集较大但评分相似度较低的用户被过滤掉。现有权重相似度虽然考虑共同评分项目集的大小，可一方面引入手动参数阈值，另一方面只降低共同评分数较小但评分相似较高的情况，未增加共同评分数较大的但评分相似较低的情况。为了更好地说明问题，我们统计了数据集中用户共同评分数的分布及其与 Pearson 相似度间的关系。

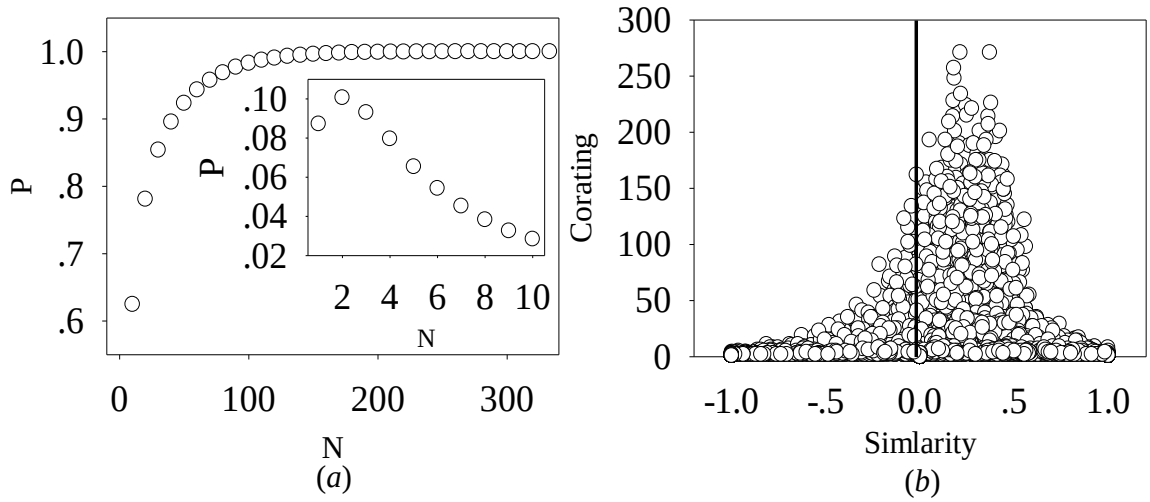


图 5-1 (*a*)为共同评分数的分布，(*b*)为相似度与共同评分的关系

图 5-1(*a*)为共同评分数分布，横坐标 N 为共同评分数，纵坐标 P 为共同评分数占所有评分数的累积比率。从中我们看到只有不到 8%的共同评分数是在 50 以上，其中共同评分数不超过 10 的占到 62.4%，图 5-1(*a*)中的子图是具体分布，其纵坐标 P 为共同评分数占所有评分数的比率，从子图中看到 30%的共同评分数不超过 3 个。因此传统相似度由于没有考虑共同评分项目集的大小，通常在较小评分项目集上得到一些不能反映用户真实兴趣相似关系的相似度值。

图 5-1(*b*)为相似度值与共同评分之间的关系，横坐标 Similarity 为相似度值，纵坐标 Corating 为共同评分数。由图可知，散点图整体向右上倾斜，说明 Pearson 正相关个数多于负相关个数，且共同评分数越多，值越大，这意味着用户共同评分越多，用户兴趣就越相似。散点图从下到上由密集到稀疏，说明大多数相似度值是在较少的共同评分上计算的，这一点与图 5-1(*a*)中的分析相吻合。特别是散点图下面的点沿着横轴向两端发散，说明在-1 和 1 附近的相似度虽然较高，但通常是由较少共同评分计算而来的。

借助图 5-1(*b*)我们很清楚地看到权重相似度没有从整体角度出发，只降低了横坐标两端由较少共同评分计算而来的相似度值，并没有考虑图中靠上方有较多共同评分且更能反映用户兴趣相似关系的相似度值，由于这些相似度值普遍较低，按照 K 近邻策略无法被选为当前用户的最近邻，为此，我们应增加这些更能反映用户兴趣相似关系的相似度值。因此本文从整体角度考虑共同评分数，提出了 Sigmoid 权重相似度，避免了传统相似度和权重相似度的不足，特别是结合 Sigmoid 函数和共同评分数加权相似度，不仅降低了依据较少共同评分的相似度值，而且还增加了较多共同评分的相似度值。

5.3.2 Sigmoid 权重相似度

Sigmoid 函数是一种常用的阈值函数，很多自然过程如学习曲线和遗忘曲线等都出现 Sigmoid 函数特征：较小初值，加速增长，加速度减少，最后趋于稳定[88]，其数学表达式见(5-1)式，

$$f(x)=\frac{2}{1+e^{-x}}-1 \tag{5-1}$$

Sigmoid 函数图象如图 5-2 所示。

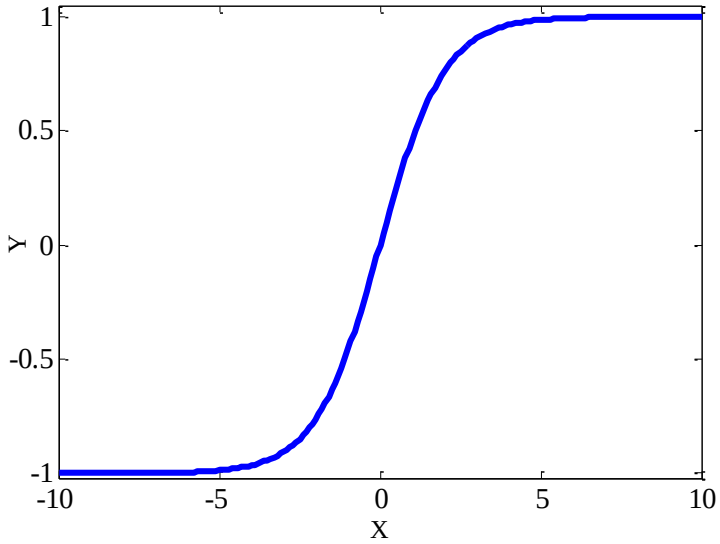


图 5-2 Sigmoid 函数图象

本文利用 Sigmoid 函数的特性，对用户间的共同评分数进行修正，以此作为相似度权重，但由于 Sigmoid 函数收敛速度快，无法有效区分共同评分数对相似度的影响，经多次实验采用对数函数来降低 Sigmoid 函数的收敛速度。对数调整后的 Sigmoid 函数表达式见(5-2)式

$$f(x)=\frac{2}{1+e^{-log_{10}(|x|)}}-1 \tag{5-2}$$

对数调整的后的函数和原始的 Sigmoid 函数图像对比见图 5-3，从图中我们可以看出，经对数调整后的 Sigmoid 函数收敛速度更慢。

用户 u_i 和 u_j 的 Sigmoid 权重相似度如下

$$sim(u_i,u_j)=(\frac{2}{1+e^{-log_{10}(|I_i\cap I_j|)}}-1)\times sim'(u_i,u_j) \tag{5-3}$$

其中 I_i，I_j 为用户 u_i 和 u_j 的已评分项目集合，$sim'(u_i,u_j)$ 为用户 u_i 和 u_j 的 Pearson 相似度。

经过 Sigmoid 函数调整后的相似度范围为

$$sim(u_i,u_j)=\begin{cases}-1, & |I_i\cap I_j|=0\\ [0,1], & |I_i\cap I_j|>0\end{cases}$$

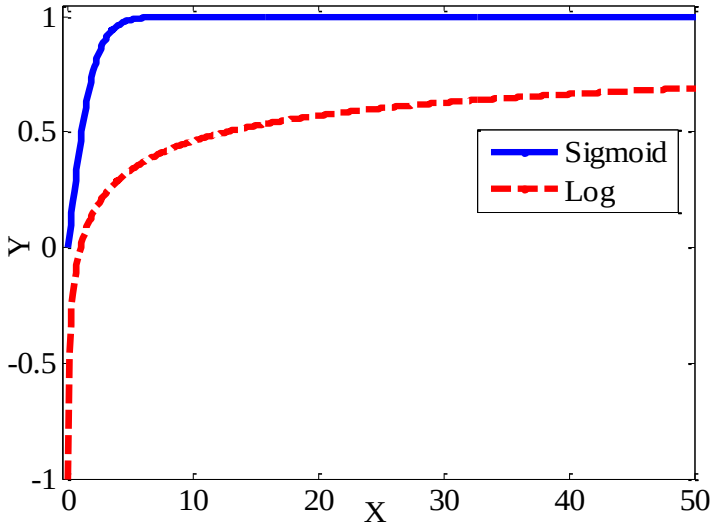


图 5-3 经对数调整后的 Sigmoid 函数

5.4 实验过程及结果分析

为了验证本文提出的 SWCF 算法的性能，本文做了四组实验将本文算法与基于 Pearson 相似度的协同过滤算法(Collaborative Filtering Based on Pearson Similarity, PBCF)及基于 Min 权重相似度的协同过滤算法(Collaborative Filtering Based on Min Weight Similarity, Min)的推荐结果分别进行比较，结果见图 5-4。

5.4.1 SWCF 与 CF 性能比较及分析

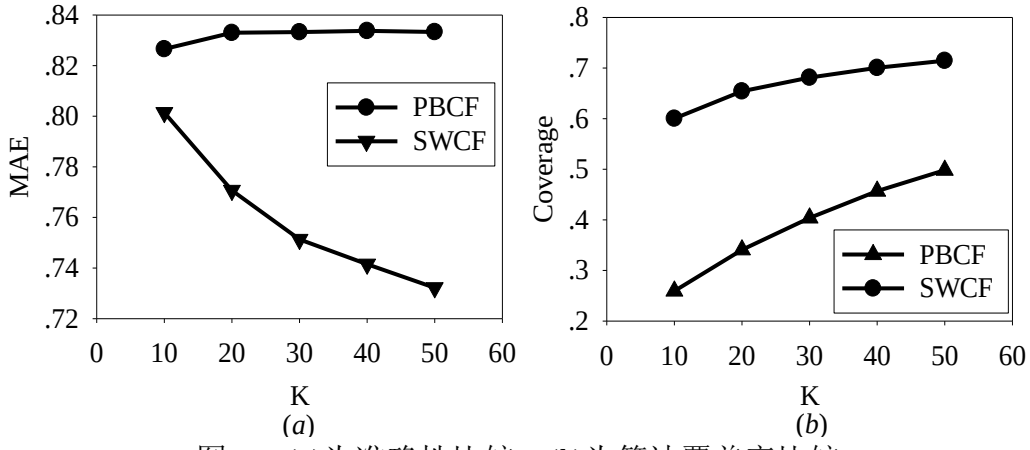


图 5-4 (*a*)为准确性比较，(*b*)为算法覆盖率比较

为了检验本文算法与传统协同过滤算法的性能，把算法 SWCF 和 PBCF 进行比较，邻居个数 K 从 10 到 50，间隔为 10，计算 MAE 和 Coverage，得到如图 5-4 (*a*)和图 5-4 (*b*)所示的 MAE 和 Coverage 随邻居个数变化的曲线图。由图 5-4 (*a*)知，本文 SWCF 算法的 MAE 一直低于 PBCF，随着邻居个数增加，算法提升幅

度也变大，平均提升幅度达到 8.71%，这表明 Sigmoid 权重相似度能提高算法的准确性。由图 5-4 (*b*)知，本文 SWCF 算法的 Coverage 一直高于 PBCF，特别是邻居集较小时，SWCF 算法的优势非常明显，这表明 Sigmoid 权重相似度能提高算法的覆盖率。

数据稀疏条件下，用户共同评分项目数较少，大大降低了传统相似度度量方法的准确性，而 Sigmoid 权重相似度由于结合了 Sigmoid 函数特征和共同评分大小，能比传统相似度方法更准确地度量用户间的兴趣相似关系，不仅提高了算法预测准确性，还提高了推荐覆盖率，因此 Sigmoid 权重相似度有助于缓解数据稀疏性问题。

5.4.2 Sigmoid 权重与 Min 权重比较及分析

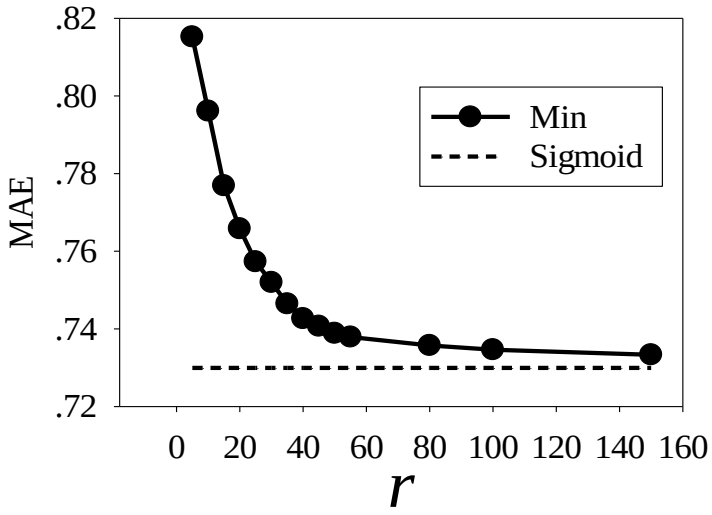


图 5-5 Min 和 Sigmoid 权重相似度比较

为了检验 Sigmoid 权重相似度与现有权重相似度的性能，把 Sigmoid 权重相似度和 Min 权重相似度进行比较，邻居个数为 45，变换阈值γ，得到 MAE 随γ变化的曲线图，如图 5-5，其中横坐标为权重阈值γ，纵坐标为 MAE。从图 5-5 中我们看到，随着γ变大，MAE 逐渐下降，但一直没有超过 Sigmoid 权重相似度，可见 Sigmoid 权重相似度的预测准确性一直高于 Min，而且γ较小时 Sigmoid 权重相似度的优势特别明显。

一方面 Sigmoid 权重相似度未引入权重阈值，不存在不稳定情况，而 Min 权重相似度受权重阈值影响，准确性不稳定；另一方面，Sigmoid 权重相似度从项目集整体出发，既降低较小共同评分上的不可信相似度，又增加较大共同评分上的相似度，弥补了 Min 权重相似度仅考虑了较小共同评分的不足。因此 Sigmoid 权重相似度能更准确地反映用户间的兴趣相似关系，从而提高准确性。

5.4.3 用户冷启动问题研究

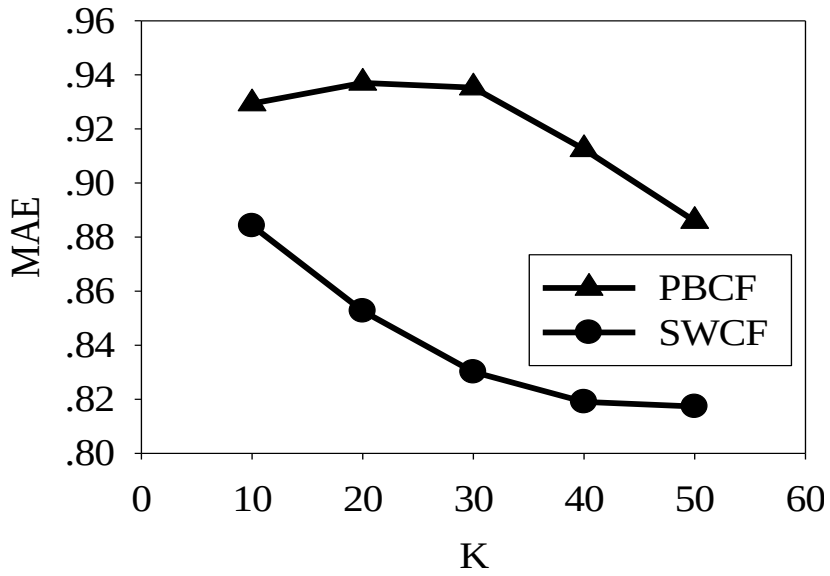


图 5-6 算法冷启动比较

用户的度表示用户选择过的产品数[2]。为了研究 Sigmoid 权重相似度对小度用户的影响，我们使用 SWCF 和 PBCF 分别计算训练集中度最小的 10 个用户的 MAE，邻居个数从 10 到 50，间隔为 10，实验结果如图 5-6 所示。由图可知 SWCF 算法的 MAE 一直低于 PBCF，这表明与 PBCF 相比，SWCF 能提高对不活跃用户的预测准确性。

新用户由于没有打分信息或者有很少的打分信息（即不活跃用户），协同过滤无法为其产生推荐，这就是一直困扰推荐系统领域的冷启动问题[4]。与 PBCF 相比，SWCF 由于结合了 Sigmoid 函数和共同评分数，能大幅度提高不活跃用户的预测准确性，平均提升幅度为 8.77%，因此 Sigmoid 权重相似度有助于缓解用户冷启动问题。

5.5 本章小结

本文在深入分析用户间共同评分分布及其与相似度关系的基础上，提出了一种结合 Sigmoid 函数和共同评分的新的权重相似度计算方法。MovieLens 数据集上的实验表明，与传统相似度相比，Sigmoid 权重相似度能同时提高预测准确性和推荐覆盖率，MAE 平均提高 8.71%，这表明 Sigmoid 权重相似度能有效缓解数据稀疏性问题。与 Min 权重相似度相比，Sigmoid 权重相似度既能提高预测准确性，又能克服参数阈值带来的不稳定性影响。进一步分析发现，Sigmoid 权重相似度对不活跃用户的 MAE 较传统协同过滤算法平均提高 8.77%，这表明 Sigmoid 权重相似度能缓解用户冷启动问题。

第六章 总结与展望

本章将总结前五章的内容，阐述本文研究中的不足之处，并展望本文将要继续研究的领域和方向。

6.1 总结

本文以推荐系统中运用最为广泛的协同过滤算法为研究对象，提出了两个改进的协同过滤算法，为推荐系统的研究和发展提供新的思路。详细来说主要工作如下：

（1）**提出了考虑负相关性信息的协同过滤算法**。首先分析了经典协同过滤算法相似度计算中存在的问题，指出了经典协同过滤算法中 Pearson 相关相似性仅考虑其正相关性信息，忽视负相关性信息，针对该问题，提出了在经典协同过滤算法的基础上考虑负相关性信息，选择最近邻时，结合由正相关性信息确定的最近邻居和由负相关性信息确定的最远邻居。实验表明了负相关信息有助于解决推荐系统中准确性和多样性两难的问题以及冷启动问题。

（2）**提出了基于 Sigmoid 权重相似度的协同过滤算法**。首先分析了传统相似度计算方法存在的问题，指出了传统相似度计算方法在共同评分项集上计算相似度，未考虑共同评分项集的大小，权重相似度虽然考虑到了共同评分项集的大小，但是引入了需要手动调节的参数，而且仅考虑了降低在较小共同评分项集上的计算而来的相似度，没有增加在较大共同评分项集上计算而来的相似度。MovieLens 数据集的实验分析，验证了上述问题的存在性。针对该问题，提出了基于 Sigmoid 权重相似度的协同过滤算法。该算法使用经改进后的 Sigmoid 函数调整后的共同评分数作为传统相似度的权重系数，实验表明了 Sigmoid 权重相似度能有效缓解数据稀疏性问题和用户冷启动问题。与 Min 权重相似度相比，Sigmoid 权重相似度既能提高预测准确性，又能克服参数阈值带来的不稳定性影响。

6.2 展望

本文总结了个性化推荐系统相关的理论和应用情况，详细分析了协同过滤算法，提炼了相关的几个改进策略。针对协同过滤算法中邻居选择和相似度计算中的存在的问题，提出了相应的改进算法，并通过实验表明改进后的算法与基准算法相比，各方面的性能均有不同程度的改善。同时本文也有很多不足，值得进一步改进之处，比如我们的算法没有考虑时间因素，没有探讨用户兴趣偏好随时间的变化规律；此外本文仅考虑基于评分的推荐算法，未考虑其他的用户行为信息，比如点击行为和社会关系信息等。个性化推荐算法还有很多其他的问题需要解决，

今后进一步的研究将主要在一下几个方面展开：

（1）社会化推荐。近年来，社交媒体快速发展为推荐系统提供了新的战场。社交媒体为推荐系统提供了新的数据，比如社会化标签、信任关系、朋友关系、社交关系和时间位置等上下文信息等。如何利用这些社交信息进行个性化推荐是值得研究的问题。

（2）实时性推荐。信息技术的发展极大地促进了信息产生和传播的速度，有时候信息的实效性要高于信息本身的价值，缺乏实效性的信息价值会大大降低。如何能够根据用户近期的行为信息，快速准确的获取最新的兴趣爱好，对推荐系统的实时性提出了较大的挑战。如何构建新的实时推荐系统架构，或者是研究新的快速算法是一个激动人心的研究方向。

（3）大数据推荐。新兴信息技术的不断涌现，使全球的数据量不断积累，学术界和工业界一致认为我们现在已经处在大数据时代。大数据对推荐系统来说，既是机遇也是挑战。推荐系统代表了互联网人工智能的发展方向，高度依赖用户行为数据，因此大数据为推荐系统的发展提供了肥沃的数据土壤；同时大数据给推荐系统带来的大挑战包括如何从海量数据中分析用户行为，如何搭建高效的推荐系统架构。另一方面如何建立高效快速的个性化推荐算法也是将来值得研究的方向。虽说数据量越大，推荐系统的准确性会提高，但我们不需要利用所有的数据进行推荐，如何从海量数据中选取适量的数据去完成符合精度要求的推荐，将是一个值得期待也颇具应用价值的研究方向。

参考文献

[1] http://www.zdnet.com/blog/btl/how-much-data-is-consumed-every-minute/80666.

[2] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19(1): 1-15.

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

[4] 周涛. 个性化推荐技术的十大挑战[J]. 程序员, 2012 (6): 107-111.

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

[6] http://www.nature.com/news/specials/bigdata/index.html.

[7] http://www.sciencemag.org/site/special/data.

[8] 王鹏, 王晶晶, 俞能海. 基于核方法的 User-Based 协同过滤推荐算法[J]. 计算机研究与发展, 2013, 50(7): 1444-1451.

[9] 郭磊, 马军, 陈竹敏. 一种信任关系强度敏感的社会化推荐算法[J]. 计算机研究与发展,2013,50(9): 1805-1813.

[10] 张振跃, 赵科科. 数据缺损矩阵低秩分解的正则化方法[J]. 中国科学: 数学, 2013 (3): 249-271.

[11] Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications[M]. Berlin:Springer Verlag, 2001: 115-153.

[12] Schafer J B, Konstan J, Riedl J. Recommender systems in E-commerce[C]. Proceedings of the 1st ACM Conference on Electronic Commerce, Denver: ACM , 1999: 158-166.

[13] Agrawal R IT, Swami. Mining associations between sets of items in large databases[C]. Proceedings of the ACM SIGMOD international Conference on Management of Data, Washington DC:ACM, 1993: 207-216.

[14] Agrawal R SR. Fast algorithms for mining association rules[C]. Proceedings of 20th Internet Conference Very Large Data Bases, Santiago,Chile,1994,1215: 487-499.

[15] 韩家炜, 堪博, 范明, 孟小峰译. 数据挖掘概念与技术[M]. 2009: 100-120.

[16] F Xu, J Budzik, Hammond J K. Mining navigation history for recommendation[C]. Proceedings of 5th International Conference Intelligent User

Interfaces, New York: ACM , 2000: 106-112.

[17] 邢东山，沈钧毅. 基于 Web 日志的因特网协作推荐系统研究[J]. 西安交通大学学报, 2002, 36(12): 1271-1274.

[18]邓爱林, 朱杨勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报 2003, 9:1-8.

[19]Lin W Y, Alvarez S A, Ruiz C. Collaborative recommendation via adaptive association rule mining[C]. Proceedings of 6th International Conference on Knowledge Discovery and Data Mining Workshop, Boston :MA, 2000.

[20]丁振国, 陈静. 基于关联规则的个性化推荐系统[J]. 计算机集成制造系统, 2003, 9(10): 891-893.

[21]Balabanovi ć M, Shoham Y. Fab:content-based collaborative recommendation[J]. Communication of the ACM, 1997(40): 66-72.

[22]Musto C. Enhanced vector space models for content-based recommender systems[C]. Proceedings of the 4th ACM conference on Recommender systems, New York :ACM , 2010: 361-364.

[23]傅京孙. 模式识别及其应用[M]. 北京:科学出版社,1983: 50-60.

[24]边肇祺. 模式识别[M]. 北京:清华人学出版社,2000: 40-50.

[25]Billsus D, Pazzani M J. Learning collaborative information filters[C]. Proceedings of ICML ,Madison: Morgan Kaufmann,1998,98:46-54.

[26]L évy P, Bonomo R. Collective intelligence: Mankind's emerging world in cyberspace[M]. USA: Perseus Publishing, 1999.

[27] Basu C, Hirsh H, Cohen W. Recommendation as classification:using social and content-based information in recommendation[C]. Proceedings of the AAAI. Mello Park: AAAI Press,1998: 714-720.

[28]Claypool M, Gokhale A, Miranda T, et al. Combining content-based and collaborative filters in an online newspaper[C]. Proceedings of the ACM SIGIR on Recommender Systems. New York: ACM Press ,1999.

[29]Pazzani M. A framework for collaborative, content-based and demographic filtering[J]. Artificial Intelligence Review ,1999,13(5): 393-408.

[30]Burke R. Hybrid systems for personalized recommendations[M]. Intelligent Techniques for Web Personalization. Berlin: Springer Heidelberg, 2005:133-152.

[31]Zhou T, Kuscsik Z, Liu J G, et al. Solving the apparent diversity-accuracy dilemma of recommender systems[J]. Proceedings of the National Academy of Sciences,

2010, 107(10): 4511-4515.

[32] Burke R. Hybrid recommender systems: survey and experiments[J]. User Modeling and User Adapted Interaction ,2002, 12(4): 331-370.

[33] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Phys Review E, 2007, 76: 046115 57.

[34] Zhou T, Jiang L L, Su R Q, et al. Effect of initial configuration on network-based recommendation[J]. Euro phys Lett, 2008, 81: 58004.

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

[36] Zhang Y C, Blattner M, Yu Y K. Heat conduction process on community networks as a recommendation model[J]. Phys Rev Letter, 2007, 99: 154301.

[37] Zhang Y C, Medo M, Ren J, et al. Recommendation model based on opinion diffusion[J]. Euro phys Letter, 2007, 80: 68003.

[38] Liu J G, Zhou T, Che H A, et al. Effect of high-order correlations to bipartite network personalized recommendations[J]. Physical A J, 2010, 389: 881-886.

[39] Liu J G, Zhou T, Wang B H, et al. Effects of user tastes on personalized recommendation[J]. International Journal of Modern Physics C , 2009,20(12): 1925-1932.

[40] 嵇晓声, 刘宴兵, 罗来明. 协同过滤中基于用户兴趣度的相似性度量方法[J]. 计算机应用,2010,10(30): 2618-2620.

[41] Herlocker J L, Konstan J A, Riedl J. Explaining collaborative filtering recommendation[C]. Proceedings of the 2000 Conference on Computer Supported Cooperative Work, Philadelphia:ACM ,2000, 241-250.

[42] Sarwar B M, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithm[C]. Proceedings of the 10th International World Wide Web Conference, Hong Kong: ACM, 2001: 285-295.

[43] 高仕龙. 基于奇异值分解的英文文本检索算法[J]. 计算机工程, 2011, 37(1): 78-80.

[44] Sarwar B M, Karypis G, Konstan J A, et al. Application of dimensionality reduction in recommender systems-a case study[R]. USA: Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[45] 李涛, 王建东, 叶飞跃, 等. 一种基于用户聚类的协同过滤推荐算法[J]. 系统

工程与电子技术, 2007, 29(7): 1177-1178.

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

[47] Wang J, Devries A P, Reinders M J T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. Proceedings of the 29th Annual Int'l ACM SIGIR. New York: ACM , 2006. 501-508.

[48] 吴湖, 王永吉, 王哲, 等. 两阶段联合聚类协同过滤算法[J]. 软件学报, 2010, 21(5): 1042-1054.

[49] 邢艳, 周勇. 基于互近邻一致性的近邻传播算法[J]. 计算机应用研究, 2012, 29(7): 2524-2526.

[50]李德毅, 刘常昱, 杜鹢, 等. 不确定性人工智能[J]. 软件学报, 2004,15(11): 1583-1594.

[51]许鹏远, 党延忠. 基于元相似度的推荐算法[J]. 计算机应用研究, 2011, 28(10): 3646-3650.

[52]Symeonidis P, Nanopoulos A, Papadopoulos A N, et al. Collaborative filtering: Fallacies and insights in measuring similarity[C]. Proceedings of the 17th European Conference on Machine Learning and 10th European Conference on Principles and the Practice of Knowledge Discovery in Databases Workshop on Web Mining. Berlin: Springer Heidelberg, 2006: 56-67.

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

[54]Kwon H, Lee T, Hong K. Improved memory-based collaborative filtering using entropy-based similarity measures[C]. Proceedings of the 2009 International Symposium on Web Information Systems and Applications, Nanchang: Academy Publisher, 2009: 29-34.

[55] Ahn H J. A hybrid collaborative filtering recommender system using a new similarity measure[C]. Proceedings of the 6th WSEAS International Conference on Applied Computer Science, Cairo: ACM ,2007(6).

[56]石珂瑞, 刘建国, 郭强, 等. 有向相似性对协同过滤推荐系统的影响研究[J]. 复杂系统与复杂性科学, 2012, 9(3): 46-49.

[57]Zhao C, Peng Q, Liu C. An improved structural equivalence weighted similarity for recommender systems[J]. Procedia Engineering, 2011, 15: 1869-1873.

[58] Zhang Q M, Shang M S, Zeng W, et al. Empirical comparison of local structural

similarity indices for collaborative filtering based recommender systems[J]. Physics Procedia, 2010, 3(5): 1887-1896.

[59] 赵琴琴, 鲁凯, 王斌. SPCF: 一种基于内存的传播式协同过滤推荐算法[J]. 计算机学报, 2013, 36(3): 671-676.

[60] Li Y, Wang B, Xu S, et al. Querytrans: Finding similar queries based on query trace graph[C]. Proceedings of Web Intelligence and Intelligent Agent Technologies, Milan: Iet Conference Publications,2009,1: 260-263.

[61] 李琳娜, 李建春, 张志平. 启发式的物品相似度传播的协同过滤算法研究[J]. 现代图书情报技术, 2013 (001): 30-35.

[62] Zhang J, Pu P. A recursive prediction algorithm for collaborative filtering recommender systems[C]. Proceedings of the 2007 ACM conference on Recommender systems. Silicon Valley: ACM, 2007: 57-64.

[63] 冷亚军, 梁昌勇, 陆青, 等. 基于近邻评分填补的协同过滤推荐算法[J]. 计算机工程, 2012, 38(21): 56-58.

[64] 宣照国, 苗静, 党延忠. 基于扩展邻居的协同过滤算法[J]. 情报学报, 2010 (003): 443-448.

[65] 贾冬艳, 张付志. 基于双重邻居选取策略的协同过滤推荐算法[J]. 计算机研究与发展, 2013, 50(5): 1076-1084.

[66] Huete J F, Fernández-Luna J M, De Campos L M, et al. Using past-prediction accuracy in recommender systems[J]. Information Sciences, 2012, 199: 78-92

[67] Bellogín A, Parapar J. Using graph partitioning techniques for neighbor selection in user-based collaborative filtering[C]. Proceedings of the 6th ACM conference on Recommender systems. Dublin: ACM, 2012: 213-216.

[68] 陈志敏, 沈洁, 赵耀. 基于相关均值的协同过滤推荐算法[J]. 计算机工程, 2009, 35(22): 53-55.

[69] 李永, 徐德智, 张勇, 等. 协作过滤算法中一种预测值判定方法的研究[J]. 小型微型计算机系统, 2008, 3(3): 469-472.

[70] 李春, 朱珍民, 高晓芳, 等. 基于邻居决策的协同过滤推荐算法[J]. 计算机工程, 2010, 36(13): 34-36.

[71] 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.

[72] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions [J]. IEEE Trans on

Knowledge and Data Engineering, 2005, 17(6): 734-749.

[73] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]. Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Berlin: Morgan Kaufmann , 1998: 43-52.

[74]朱丽中, 徐秀娟, 刘宇. 基于项目和信任的协同过滤推荐算法[J]. 计算机工程,2013, 39(1):58-63.

[75]Wang J, Yin J. Enhancing accuracy of User-based Collaborative Filtering recommendation algorithm in social network[C]. Proceedings of International Conference on System Science, Engineering Design and Manufacturing Informatization, Chengdu: IEEE, 2012, 1: 142-145.

[76]孙小华, 陈洪, 孔繁胜. 在协同过滤中结合奇异值分解与最近邻方法[J]. 计算机应用研究, 2006, 23(9): 206-208.

[77] 邹永贵, 望靖, 刘兆宏 ,等. 基于项目之间相似性的兴趣点推荐方法[J]. 计算机应用研究,2012,29(1):116-118.

[78]罗辛, 欧阳元新, 熊璋, 等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报,2010 ,33(8): 1437-1445.

[79] 黄创光, 印鉴, 汪静, 等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报,2010,33(8): 1369-1377.

[80]张尧, 冯玉强. 协同过滤推荐中基于用户分类的邻居选择方法[J]. 计算机应用研究,2012, 29(11): 4216-4219.

[81]Zeng W, Shang M S, Zhang Q M, et al. Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation?[J]. International Journal of Modern Physics C, 2010, 21(10): 1217-1227.

[82] 刘建国, 周涛, 郭强, 等. 个性化推荐系统评价方法综述[J]. 复杂系统与复杂性科学,2009,6(003): 1-10.

[83] Herlocker J, Konstan J A, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms[J]. Information Retrieval ,2002,5(4): 282-310.

[84] Mclaughlin M R, Herlocker J L. A collaborative filtering algorithm and evaluation metric that accurately model the user experience[C]. Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield: ACM, 2004: 329-336.

[85] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for

collaborative filtering of netnews[C]. Proceedings of the 1994 ACM conference on Computer supported cooperative work, New York: ACM, 1994: 175-186.

[86] Candillier L, Meyer F, Fessant F. Designing specific weighted similarity measures to improve collaborative filtering systems[C]. Proceedings of the Industrial Conference on Data Mining, Berlin: Springer Verlag, 2008, 50(77): 242-255.

[87] Bell R, Koren Y, Volinsky C. Modeling relationships at multiple scales to improve accuracy of large recommender systems[C]. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, USA: ACM, 2007: 95-104.

[88] 方耀宁, 郭云飞, 扈红超, 等. 一种基于 Sigmoid 函数的改进协同过滤推荐算法[J]. 计算机应用研究, 2013, 30(6): 1688-1691.

在读期间公开发表的论文和承担科研项目及取得成果

一、论文

1. 郭强, **周继平**, 郭迎迎, 胡兆龙. 考虑负相关性信息的协同过滤算法研究[J].计算机应用研究, 2013.30(12):3540-3542.

2. **周继平**, 郭强, 刘建国,胡兆龙. 协同过滤算法中一种新的权重相似度[J].计算机应用研究(审稿中).

二、参与的科研项目

1. 教育部科学技术研究重点项目,动态网络上的信息传播规律与引导策略研究(211057), 2012.01-2013.12, 参与人.

致 谢

丹桂飘香之际，两年半的研究生时光即将结束，我的硕士毕业论文也即将告一段落。回首上理硕士求学的这两年半，有科学研究过程中所遇到的迷茫和艰辛，更有点滴收获后的欣喜。我内心深知该篇硕士论文的顺利完成，得益于众多师长、同学、亲人和朋友的支持和鼓励，感激之情，片纸难表。

衷心感谢我的导师郭强副教授。郭老师学识渊博，治学严谨，为人谦和，是我学习的榜样。郭老师不仅是我学术上的导师，更是我生活和工作上的导师。郭老师在学术上给予了我很大的帮助，在生活和工作上给予了我无私的关怀。郭老师教诲我做人做事的道理，一次次难忘的长谈都永远铭记在学生心中。正是郭老师的严格要求和悉心指导，我才能取得一点点小成绩，才能顺利完成毕业论文撰写。在此，学生向郭老师表示深深的感谢和崇高的敬意。

衷心感谢刘建国教授无私的培养、帮助、鼓励和指导。刘老师思维敏捷、思想深邃，严谨练达、细致缜密的治学态度使我受益匪浅。刘老师是我学术研究的引导者，论文中的每一件话，每一副图片都经过刘老师认真细致的修改。刘老师不仅培养了我克服学术困难的决心，更培养了我面对困难勇往直前的积极进取之心。刘老师对我前瞻性指导和心灵上的帮助，让我受益终身。

衷心感谢王恒山教授悉心指导和无私帮助。王老师胸怀广阔、平易近人的长者之风使我如浴春风，见识过人、治学严谨让我受益匪浅。王老师是我学术研究的启蒙恩师，帮我带入了科学研究的世界，锻炼了我科学研究的基本思维。王老师为我创造了非常难得的进入复杂系统研究中心的学习条件，让我在后来的学习中，收获丰厚。

感谢复杂系统研究中增加帮助过我的兄弟姐妹们，他们是石珂瑞、冷瑞、胡兆龙、邵凤、任卓明、李洋、刘新惠、杨光勇、侯磊、李旭东。他们曾多次对我研究的课题给予了非常大的帮助，让我收获颇丰。

感谢父母的支持和理解，感谢他们含辛茹苦的培养和教诲。感谢我的同学和学长学姐在研究生期间对我的帮助，他们是郑翠翠、王勇、王艳灵、高春昌、盛晓华。

支持和关心我的人还有很多，在此不一一列举，最后向所有支持和帮助过我的老师、同学和朋友表示最深的敬意和诚挚的谢意。

周继平

二 O 一三年十二月于上海理工大学